z/OS

DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support

SC35-0426-01

z/OS

IBM

DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support

SC35-0426-01

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 529.

Second Edition, March 2002

This edition applies to Version 1 Release 3 of z/OS™ (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC35-0426-00.

Order publications through your IBM® representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
RCF Processing, Department M86/050
5600 Cottle Road
San Jose, CA 95193-0001
United States of America

IBMLINK from US: STARPUBS at SJEVM5
IBMLINK from Canada: STARPUBS at TORIBM
IBM Mail Exchange: USIB3VVD at IBMMAIL
Internet: starpubs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

About This Book

This book introduces OAM and explains how to do the following tasks:
- Plan for the installation of OAM
- Install OAM
- Customize OAM
- Administer OAM
- Operate OAM

This book is for the system programmers, storage administrators, and system operators who perform these tasks.

Required Product Knowledge

To use this book effectively, you should be familiar with:
- IBM DATABASE 2™ (DB2®)
- Data Facility Storage Management System (DFSMS)
- Interactive Storage Management Facility (ISMF)
- Customer Information Control System (CICS®)—optional, depending on your installation
- Information Management System (IMS™)—optional, depending on your installation

Referenced Publications

The following publications are referenced in this book, or are useful in understanding and applying the material presented:

Table 1. Referenced Publications

Publication Title	Order Number
IBM 3480 Magnetic Tape Subsystem Introduction	GA32-0041
IBM 3490 Magnetic Tape Subsystem Models A01, A02, A10, A20, B02, B04, B20, and B40 Introduction	GA32-0125
IBM 3490 Magnetic Tape Subsystem Enhanced Capability Models C10, C11, and C22 Introduction	GA32-0217
IBM TotalStorage Enterprise Tape Library (3494) Introduction and Planning Guide	GA32-0448
IBM TotalStorage Enterprise Automated Tape Library (3495) Introduction	GA32-0234
IBM TotalStorage Enterprise High Performance Tape System 3590 Introduction and Planning Guide	GA32-0329
IBM 3995 Optical Library Dataserver Products: Introduction and Planning Guide	GA32-0121
IBM 3995 Optical Library Dataserver Products: Optical Disk Cartridge Requirements 130 mm 1024 Bytes/Sector	GA32-0146
IBM 3995 ESA/370 and ESA/390 Optical Library Dataserver: Reference Models 132, 131, 112, and 111	GA32-0145
IBM 3995 ESA/370 and ESA/390 Optical Library Dataserver: Operator's Guide Models 133, 132, 131, 113, 112, and 111	GA32-0122

Table 1. Referenced Publications (continued)

Publication Title	Order Number
IBM 3995 Optical Library Dataserver: Operator's Guide for C-Series Model	GA32-0352
IBM 3995 Optical Library Dataserver: Introduction and Planning Guide for C-Series Models	GA32-0350
IBM 3995 ESA/370 and ESA/390 Optical Library Dataserver: Reference for C-Series Models	GA32-0351
IBM 9246/9247 Reference Summary for MVS/ESA	GX35-5041
CICS Application Programming Reference	SC34-5703
CICS Customization Guide	SC34-5706
CICS Transaction Server for OS/390 Installation Guide	GC34-5697
CICS Operations and Utilities Guide	SC34-5717
CICS Resource Definition Guide	SC34-5722
CICS DB2 Guide	SC34-5707
DB2 Administration Guide	SC26-8957
DB2 Application Programming and SQL Guide	SC26-8958
DB2 Call Level Interface Guide and Reference	SC26-8959
DB2 Command Reference	SC26-8960
DB2 SQL Reference	SC26-8966
DB2 Utility Guide and Reference	SC26-8967
DB2 What's New	GC26-8971
DB2 Messages and Codes	GC26-8979
DB2 Diagnosis Guide and Reference	LY27-9659
z/OS DFSMS Access Method Services	SC26-7394
z/OS DFSMSdfp Diagnosis Guide	GY27-7617
z/OS DFSMSdfp Diagnosis Reference	GY27-7618
z/OS DFSMS Introduction	SC26-7397
z/OS DFSMS: Managing Catalogs	SC26-7409
z/OS DFSMS OAM Application Programmer's Reference	SC35-0425
z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	SC35-0427
z/OS DFSMS: Using the Interactive Storage Management Facility	SC26-7411
IMS/ESA Application Programming: EXEC DLI Commands for CICS and IMS	SC26-8726
z/OS MVS Initialization and Tuning Reference	SA22-7592
z/OS Parallel Sysplex Application Migration	SC22-7662
z/OS Parallel Sysplex Overview	SA22-7661
z/OS MVS Setting Up a Sysplex	SA22-7625
OS/390 MVS Parallel Sysplex Configuration, Volume 1: Overview	SG24-2075
RMF User's Guide	SC33-7990
z/OS DFSMS: Implementing System-Managed Storage	SC26-7407
z/OS HCD User's Guide	SC33-7988

Table 1. Referenced Publications (continued)

Publication Title	Order Number
z/OS MVS System Commands	SA22-7627
z/OS MVS System Messages, Vol 4	SA22-7634
z/OS Program Directory	GI10-0669

Accessing z/OS DFSMS Books on the Internet

In addition to making softcopy books available on CD-ROM, IBM provides access to unlicensed z/OS softcopy books on the Internet. To find z/OS books on the Internet, first go to the z/OS home page: http://www.ibm.com/servers/eserver/zseries/zos

From this Web site, you can link directly to the z/OS softcopy books by selecting the Library icon. You can also link to IBM Direct to order hardcopy books.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages, system abends, and some codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html`

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, `IAT1836` or `IAT*`). You can select a specific release to narrow your search. You can also download code from the *IBM Online Library Omnibus Edition z/OS Collection*, SK2T-6700 and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *IBM Online Library Omnibus Edition z/OS Collection*, SK2T-6700 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html.
2. Click the **News** button.
3. Scroll to **Download LookAt Code for TSO and VM.**
4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the **lookat.me** file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt prompts you to choose which book to open.

Accessing Licensed Books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

`http://www.ibm.com/servers/resourcelink`

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:

`http://www.ibm.com/servers/resourcelink`

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link user ID and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed, you will receive an e-mail confirmation.

Note: You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSMS documentation, use one of the following methods to provide feedback:

- Send your comments by e-mail to:
  - IBMLink™ from US: starpubs@us.ibm.com
  - IBMLink from Canada: STARPUBS at TORIBM
  - IBM Mail Exchange: USIB3VVD at IBMMAIL
  - Internet: starpubs@us.ibm.com

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, Department 61C, 9000 South Rita Road, Tucson, Arizona, 85744-0002.

How to Read Syntax Diagrams

There is one basic rule for reading the syntax diagrams: Follow only one line at a time from the beginning to the end and code everything you encounter on that line.

The following rules apply to the conventions that are used in the syntax diagrams for all the OAM commands:

- Read the syntax diagrams from left to right and from top to bottom.
- Each syntax diagram begins with a double arrowhead (►►) and ends with opposing arrows (►◄).
- An arrow (─►) at the end of a line indicates that the syntax continues on the next line. A continuation line begins with an arrow (►─).
- Commands and keywords are shown in uppercase letters.
- Some commands and keywords have alternative abbreviations; these appear as part of the stack for that command or keyword. For example, the alternative abbreviation for **DISPLAY** is **D**.

```
►►──┬─DISPLAY─┬──────────────────────────────────────────────────►◄
    └─D───────┘
```

- Where you can choose from two or more keywords, the choices are stacked one above the other. If one choice within the stack lies on the main path, a keyword is required, and you must choose one. In the following example you must choose either **DETAIL** or **STATUS**.

```
►►──┬─DETAIL─┬───────────────────────────────────────────────────►◄
    └─STATUS─┘
```

- If a stack is placed below the main path, a keyword is optional, and you can choose one or none. In the following example, **PURGE**, **KEEP**, and **LOCATION** are optional keywords. You can choose any one of the three.

```
►►──┬──────────┬─────────────────────────────────────────────────►◄
    ├─PURGE────┤
    ├─KEEP─────┤
    └─LOCATION─┘
```

- Where you can choose from two or more keywords and one of the keywords appears above the main path, that keyword is the default. You may choose one or the other of the keywords, but if none is entered, the default keyword is automatically selected. In the following example you may choose either **DETAIL** or **STATUS**. If neither is chosen, **STATUS** is automatically selected.

```
      ┌STATUS┐
►►────┴DETAIL┴──────────────────────────────────────────────────►◄
```

- Words or names in italicized, lowercase letters represent information that you supply. The values of these variables may change depending on the items to which they refer. For example, *volser* refers to the serial number of a volume, while *storgrp_name* refers to the name of a storage group.
- You must provide all items enclosed in parentheses ( ). You must include the parentheses. In the following example, you must supply the volume serial number (*volser*) and it must be enclosed in parentheses.

```
►►──┬DISPLAY┬──SMS,──┬VOLUME┬──(volser)──────────────────────►◄
    └D──────┘        └VOL───┘
```

You would code this as follows:

D SMS,VOL(*volser***)**

The variable *volser* is the serial number of the volume you wish to display.

- The repeat symbol shown below indicates that you can specify keywords and variables more than once. The repeat symbol appears above the keywords and variables that can be repeated. For example, when a comma appears in the repeat symbol, you must separate repeated keywords or variables with a comma.

In the following example, you may specify the *library_name* and one or more system identification numbers (*system_id*) that are separated by commas. You must enclose the name of the library and all of the system IDs in parentheses.

```
►►──(library_name──┬──────────────────┬─)─────────────────────►◄
                   │    ┌,─────────┐  │
                   │    ▼          │  │
                   └,───system_id──┴──┘
```

You would code this as follows:

(*library_name***,** *system_id***,** *system_id***,** *system_id***)**

The variable *library_name* is the name of the library you are working with, and *system_id* names three different instances of system identification numbers.

Summary of Changes

This book contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You may notice changes in the style and structure of some content in this book—for example, headings that use uppercase for the first letter of initial words only or procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our books.

Summary of Changes for SC35-0426-01 z/OS Version 1 Release 3

This book contains information previously presented in *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* (SC35-0426-00).

The following sections summarize the changes to that information.

New Information

This edition includes the following new information:

- The subsection “SETOSMC Statement for OSMC Support” on page 7 has been added to describe how you may associate an Object storage group with an Object Backup storage group that stores the first or second backup copies of objects.
- Three new figures were added to depict the process of storing primary, first backup, and second backup copies of objects. These are Figure 4 on page 18, Figure 5 on page 19, and Figure 6 on page 20.
- Information was added to “OAMplex Restrictions” on page 5 to clarify use of optical libraries defined in a source control data set (SCDS) as well as use of 9246 optical libraries and 9247 optical drives in an OAMplex.
- Information has been added to “SMS Constructs and ACS Routines” on page 10 to delineate the SMS management policy parameters.
- Information has been added to “Object, Object Backup, and Tape Storage Groups” on page 11 to describe the second backup copy of objects function.
- Information has been added to “Defining an Object or Object Backup Storage Group” on page 12 to describe the use of ISMF in defining Object and Object Backup storage groups.
- Information has been added to “Assigning Object Backup Storage Groups” on page 13 to describe how to direct OAM to create first and second backup copies of objects.
- Information has been added to “Determining Which Media to Use for Backup Copies” on page 14 to describe the ways in which OAM schedules writes of first and second backup copies of objects.
- The number of backup versions field has been added to Table 2 on page 22.
- Information has been added to “Determining Data Class During Scratch Tape Allocation” on page 22 to describe the second backup copy of an object’s DATACLASS specification for the associated tape support data set.
- Information has been added to “Determining Class Transition and Backup Requirements” on page 52 to describe how the management class parameters determine the assignment of first and second backup copies to the appropriate Object Backup storage groups.

- Special circumstances for working with the DEMOUNTWAITTIME parameter were added to “DEMOUNTWAITTIME” on page 93 and “DEMOUNTWAITTIME” on page 100.
- Information has been added to “Installation and Migration Checklist” on page 79 to include the steps that are needed to migrate to the new installation (z/OS V1R3) and to take advantage of the second backup copy of objects capability. Notes were also added to the checklist to guide DASD-only installations through migration operations.
- A new section, “Displaying Volumes that Have LOSTFLAG Set” on page 301, has been added to describe the added functionality of displaying optical and tape volumes that have LOSTFLAG set. Command syntax and display output are included.

Deleted Information

The following information was deleted from this edition:

- “Defining Optical Libraries” in “Chapter 1. Understanding the Object Access Method” on page 1 has been deleted.
- “Determining DRIVE STARTUP THRESHOLD for Object Backup Storage Group” in “Chapter 3. Migrating, Installing, and Customizing OAM” on page 77 has been deleted. The pertinent information in this section was combined with the information in “Improving Performance with Low DRIVE STARTUP THRESHOLD Value” on page 130.
- The figure “Conceptual Overview of Storing a Primary, Backup, or Secondary Backup Object to Tape” in “Chapter 1. Understanding the Object Access Method” on page 1 has been deleted.
- The subsection entitled “Creating or Updating the CBROAM*xx* PARMLIB Member with SETOAM Statements” in “Chapter 3. Migrating, Installing, and Customizing OAM” on page 77 has been deleted. The information from this subsection has been rolled into “SETOAM Statements for Object Tape Support” on page 91.

Changed Information

The following information was changed in this edition:

- The figure “WORM and Rewritable Media Characteristics” was broken out into two separate figures: Figure 8 on page 28 and Figure 9 on page 28.
- Three columns were deleted (“120”, “130”, and “130/SPE”) from and two columns were added (“OS/390 V2R10” and “z/OS V1R3”) to Table 19 on page 78. A new row, “Creating and Binding DB2 Packages”, was also added to this table.

Chapter 1. Understanding the Object Access Method

The Object Access Method (OAM) is an access method supporting a class of data referred to as objects. An *object* is a named stream of bytes. The content, format, and structure of that byte stream are unknown to OAM. There are no restrictions on the data in an object. For example, an object can be a compressed scanned image or coded data. Objects are different from data sets handled by existing access methods. The following characteristics distinguish them from traditional data sets:

- **Lack of record orientation.** There is no concept of individual records within an object.
- **Broad range of size.** An object can contain less than one kilobyte or up to 50 megabytes of data.
- **Volume.** Objects are usually much smaller than data sets; however, they are more numerous and consume vast amounts of external storage.
- **Varying access-time requirements.** Reference patterns for objects change over time or cyclically, allowing less-critical objects to be placed on lower-cost slower devices or media.

Describing Collections

A *collection* is a group of objects typically having similar performance, availability, backup, retention, and class transition characteristics. A collection is used to catalog a large number of objects, which, if cataloged separately, could require an extremely large catalog. Every object must be assigned to a collection. Object names within a collection must be unique; however, the same object name can be used in multiple collections. Each collection belongs to one and only one Object storage group. Each storage group can contain from one to many collections.

Providing an Application Interface (OSREQ Macro)

OAM provides an application programming interface known as the OSREQ macro which sets up (ACCESS) the environment for a user to use OSREQ to change, store, retrieve, delete, and query information about an object, and then releases (UNACCESS) the resources required for this macro when they are no longer needed. OAM includes the functions necessary to manage the objects after they are stored. For more detailed information on the OSREQ macro, refer to *z/OS DFSMS OAM Application Programmer's Reference*.

Using System-Managed Storage for Data and Space Management

OAM is a component of DFSMSdfp, the base for the Storage Management Subsystem (SMS) of DFSMS. OAM uses the concepts of system-managed storage, introduced by SMS, which provide functions for data and space management. The Storage Management Subsystem provides the following benefits:

- Facilitates the management of storage growth
- Improves the use of storage space
- Reduces the effort of device conversion and coexistence
- Provides centralized control of external storage
- Exploits the capabilities of available hardware

SMS is extended to include the definition of a storage hierarchy for objects and of the parameters for managing those objects. OAM uses the SMS-supplied hierarchy definition and management parameters to place user-accessible objects anywhere in the storage hierarchy.

The object storage hierarchy can consist of:
- Direct access storage device (DASD)
- Tape volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)
- Optical volumes inside a library device (SMS-managed, library-resident optical volumes), and optical volumes outside of a library device (SMS-managed, shelf-resident optical volumes)

Moving Objects Throughout the Hierarchy

During the storage management cycle, OAM determines whether the primary copy of an object is correctly positioned in the OAM storage hierarchy. See "Media Selection for Object Storage" on page 16 for information regarding the criteria used in placing objects onto the appropriate media type. If the object is not correctly positioned in the OAM storage hierarchy, the primary copy of the object is moved to the correct storage medium. This means one of following medium transitions can be performed for the primary copy of an object:
- DASD to tape
- DASD to optical
- Tape to DASD
- Tape to optical
- Optical to DASD
- Optical to tape

The location of an object in the hierarchy is unknown to the user of the programming interface. Device-dependent information is not required of the user; for example, there are no JCL DD statements and no considerations for device geometry, such as track size.

Hardware and Software Interaction with OAM

Figure 1 on page 3 shows the hardware and software in a typical OAM object support environment and illustrates the possible interactions. Notice that OAM object support is closely tied to Storage Management Subsystem (SMS) and DB2. OAM also interacts with the OAM thread isolation support (OTIS). OTIS is an OAM subsystem that provides OAM-to-DB2 functions that use a different thread to DB2 than the application program thread. Applications can use OSREQ, the application programming interface, to interact with OAM. Applications can also communicate directly with DB2. Each application is responsible for synchronizing its DB2 databases, whether the operation is generated by the application or by OAM. For more information on the application programming interface, refer to the *z/OS DFSMS OAM Application Programmer's Reference*.

Note: Due to the various device types (tape and optical) OAM supports, tape cartridge and optical disk symbols are used within the artwork in this publication to depict tape and optical storage libraries and devices. For a detailed list of the devices OAM supports, see "Hardware" on page 55.

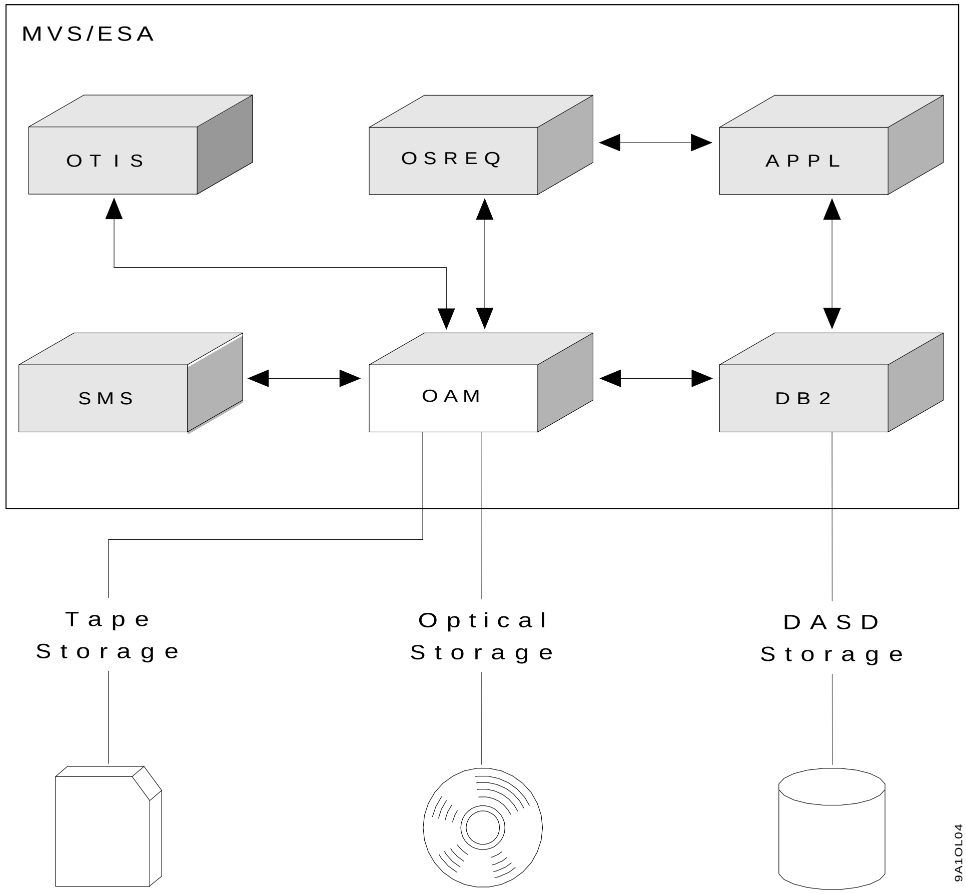

Figure 1. OAM Interactions with Software and Hardware

Parallel Sysplex™ Support

OAM supports the Parallel Sysplex environment. A Parallel Sysplex links many systems together and provides multisystem data sharing through the use of the cross-system coupling facility (XCF) component of MVS/ESA™. XCF services allow authorized applications on one system to communicate with applications on the same system or on other systems. They also allow data to be shared between the applications on these systems. The system linking and multisystem data sharing makes the sysplex platform ideal for parallel processing. With this support, objects can be accessed from all instances of OAM and from optical hardware within the sysplex, and transactions can be processed more efficiently.

For more information regarding a Parallel Sysplex, refer to:
- *z/OS Parallel Sysplex Overview*
- *z/OS Parallel Sysplex Application Migration*

OAMplex

OAM supports the concept of an **OAMplex**. An OAMplex consists of one or more instances of OAM (at a minimum level of DFSMS 1.5.0) running on systems that are part of a Parallel Sysplex. An OAMplex has a one-to-one correlation to an XCF group in a Parallel Sysplex. The XCF group associated with an OAMplex is the XCF group joined by instances of OAM address spaces, running on separate systems in a Parallel Sysplex, sharing a common OAM database in a DB2 sharing group. Each instance of OAM is a member of the same XCF group. Also, the DB2 subsystems connected to these instances of OAM belong to the same DB2 data sharing group. The instances of OAM belonging to the same XCF group are the instances of OAM that are able to communicate with each other through the services of the XCF component. The DB2 data sharing group allows the DB2 database information (OCDB, OAMADMIN, and object databases) accessed by the DB2 subsystems to be shared among OAMs belonging to the OAMplex. When different OAMs sharing a common database on DB2 join an XCF group to become an OAMplex, all object data and configuration information is known to all instances of OAM in the OAMplex. This makes it possible for any object, regardless of which OAM stored the object, to be retrieved by any instance of OAM in the OAMplex.

Note: In a Parallel Sysplex, there can only be one OAM XCF group (OAMplex) sharing a single common DB2 database. All instances of OAM running in XCF mode in a Parallel Sysplex sharing a common DB2 database **must** join the same XCF group. Also, all instances of OAM in an OAMplex must share a common catalog where the OAM collection names are defined. There can be multiple OAMplexes within the Parallel Sysplex, but each OAMplex must use a different shared DB2 database. No two OAMplexes can share the same DB2 database. Additionally, OAMs that are not in XCF mode cannot share the DB2 database.

OAM uses the XCF messaging facilities to communicate between systems, synchronize resource information, and coordinate where transactions should be processed.

With DFSMS 1.5.0 support, OAM can be running in XCF mode (in an OAMplex), or non-XCF mode (not in an OAMplex). When OAM is running as part of an OAMplex on a system in a Parallel Sysplex, that instance of OAM must be initialized with a CBROAM*xx* PARMLIB member, which specifies an XCF member name and an XCF group name using the OAMXCF statement.

As long as all instances of OAM involved with the transaction belong to the same OAMplex, any OAM object can be retrieved from any z/OS system in a Parallel Sysplex. This is allowed regardless of which OAM in the sysplex stored the object or on which medium (3995 optical, tape, or DASD) the object resides.

Transaction shipping is used to send and receive requests between OAMs within the OAMplex. In an optical environment, transaction shipping allows any OAM in a Parallel Sysplex to write objects to, retrieve objects from, or delete objects from any 3995 optical volume in the Parallel Sysplex regardless of which OAM in the sysplex controls the resource required (optical volume, optical library, or optical drive) as long as each instance of OAM is part of the same OAMplex and shares the same configuration. Requests to read data from or write data to 3995 optical volumes that reside in a 3995 optical library being managed by a different OAM on a separate z/OS system in the same Parallel Sysplex are serviced by sending the request (using XCF) to the OAM running on the z/OS system that controls the 3995 optical library dataserver. This is possible only as long as both the requesting and responding OAMs are members of the same OAMplex. 3995 optical library dataservers are still controlled and managed by a single OAM running on a single z/OS system. In the event of a system failure, control of a 3995 optical library dataserver can be switched to another OAM running on another z/OS system in the same OAMplex.

Note: When there are multiple OAMplexes within an MVS Parallel Sysplex, each OAMplex must have a unique set of OAM resources (optical devices and media for object storage) defined in its configuration.

In an object tape environment, the basic concept of transaction shipping (sending transaction requests between OAMs within the same OAMplex for processing) still pertains. However, MVS dynamic allocation is used to handle the required tape resource allocation, because OAM does not control tape resources. Tape resources are allocated as needed and only for the time required for their use.

For object tape processing, tape drives must be available to any OAM in an OAMplex where a tape request can need to be processed. Tape transactions are shipped across systems *only* when the requested tape volume for a *retrieve* request is currently allocated and mounted on a tape drive that is in use by another OAM in the OAMplex. It is otherwise expected that there are available tape drives on the system where the request originated to satisfy the request. Tape *write* requests are not sent across systems for processing.

OAMplex Restrictions

There are some restrictions with an OAMplex that you should keep in mind:

- Any instance of OAM running on a system in a Parallel Sysplex not running in XCF mode (not part of an XCF group) cannot share any resources (optical libraries, optical drives, optical volumes, or tape volumes for object data) that are owned by another instance of OAM.
- Any OAM not running in XCF mode cannot share its DB2 databases with any other instances of OAM.
- Optical libraries that are defined in a source control data set (SCDS) as connected to a system where OAM is not running in XCF mode must be logically connected to only that system.
- When you define optical libraries in an SCDS as logically connected to multiple systems, all instances of OAM on those systems must be part of a single OAMplex.
- If an OAM DB2 database is being shared, the catalog used for the OAM collection names must also be shared.

It is important that these restrictions be implemented and adhered to. OAM cannot detect or prohibit processing that does not conform to these standards, so unexpected results can occur if these restrictions are not strictly enforced.

Note: OAM does not support 9246 optical libraries or 9247 optical drives in an OAMplex. Object data stored on Filenet optical media (shelf-resident or in 9246 optical libraries) can only be retrieved from the OAM to which it is connected and online, regardless if that OAM is part of an OAMplex. The object data cannot be accessed by any other OAM within an OAMplex. If it is necessary for this object data to be accessed by more than one OAM within the OAMplex, the data must be moved to another supported media type. This way, it can be retrieved and be accessible to the other instances of OAM within the OAMplex. Otherwise, if any other system or OAM (other than the one that the 9246 optical library is connected to) attempts to retrieve an object that resides on a volume in a 9246 library, OAM returns a volume unknown error, and the attempt fails.

OAM Components

The functions of OAM are performed by its three components, as illustrated in Figure 2:

1. The **Object Storage and Retrieval (OSR)** component is an application programming interface for OAM. Applications operating in the Customer Information Control System (CICS), Information Management System (IMS™), TSO, and MVS/ESA environments use this interface to store, retrieve, query, and delete objects, and to change information about objects. OSR stores the objects in the storage hierarchy and maintains the information about these objects in DB2 databases. OSR functions, invoked through the application programming interface, require the OAM thread isolation support (OTIS) application for administrative processing.

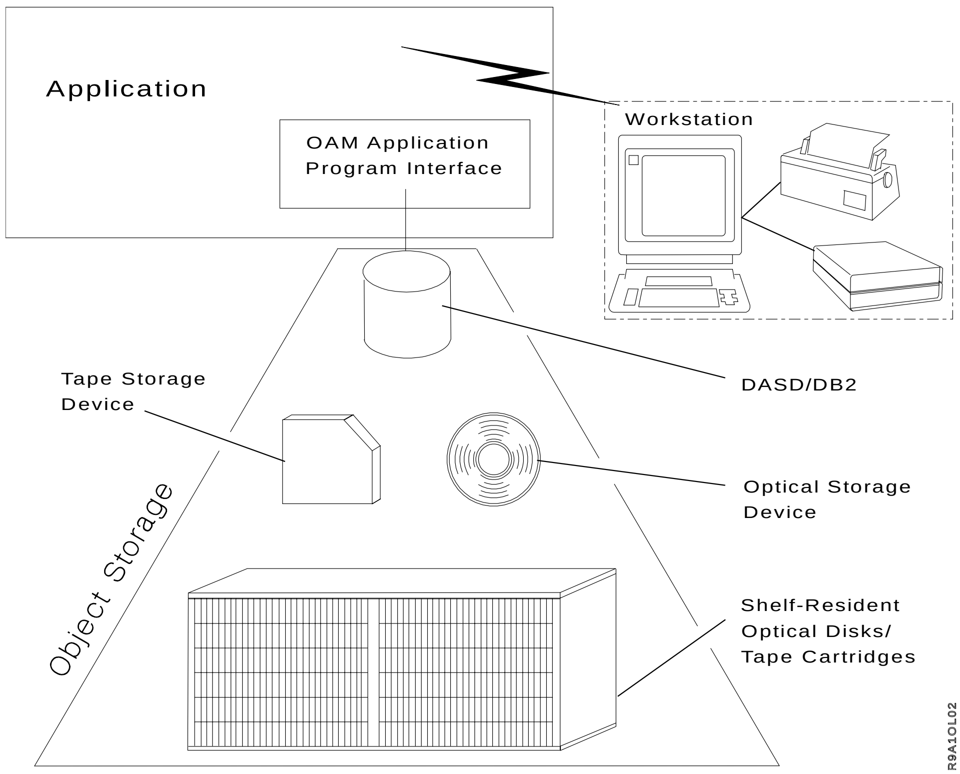

Figure 2. Application Illustration

2. The **Library Control System (LCS)** component writes and reads objects on tape and optical disk storage. It also manipulates the volumes on which the objects reside. The LCS component controls the usage of optical hardware resources that are attached to the system.

   Note: This manual deals with the data management of tape volumes which contain objects, not the actual library management of tape library dataservers. For more information on how LCS controls the library management for tape library dataservers (automated and manual), refer to the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

3. The **OAM Storage Management Component (OSMC)** determines where objects should be stored in the OAM storage hierarchy, manages object movement within the object storage hierarchy, manages expiration attributes

that are based on the installation storage management policy that is defined through SMS, and creates the requested backup copies of the objects. OSMC also provides object and volume recovery functions.

CBROAM*xx* PARMLIB Member Statements

The CBROAM*xx* PARMLIB member establishes the environment under which OAM runs. Storage administrators can customize the CBROAM*xx* PARMLIB member by updating it with statements that alter the operating environment independently of ISMF and SMS. The statements include SETOAM, SETOPT, SETOSMC, and OAMXCF. Once you have updated the CBROAM*xx* PARMLIB member with one or more of these statements, you must restart OAM.

SETOAM Statement for Object Tape Support

The CBROAM*xx* PARMLIB member contains one or more SETOAM statements. These statements contain keywords that are used to tailor the object tape support. The information that is specified on these statements can supplement or override information that was previously specified when the applicable Object or Object Backup storage group was defined using ISMF. Some keywords apply to all of the Object or Object Backup storage groups that use the object tape support, and others apply only to the group for which they have been explicitly specified. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for information on changing the SETOAM values dynamically, and on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

SETOPT Statement for Optical Support

The CBROAM*xx* PARMLIB member contains one or more SETOPT statements. The SETOPT statement and its associated keywords are similar to the SETOAM statement in that they define rules at global and storage group levels that are used by OAM to administer optical support.

The OPTICALREINITMODE keyword for this statement can be used at the OAM global level or at a specific storage group level. All other keywords must be used at the OAM global level. For more information on changing the SETOPT values dynamically, or defining the values when the CBROAM*xx* PARMLIB member is not used at initialization, see "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311.

SETOSMC Statement for OSMC Support

The CBROAM*xx* PARMLIB member contains one or more SETOSMC statements. The SETOSMC statement and its associated keywords determine the valid values of settings for OSMC processing. They associate an Object storage group with the Object Backup storage group that stores the first or second backup copies of objects. The SETOSMC statement determines which Object Backup storage groups contain the first and second copies of the objects that are associated with an Object storage group. If you do not provide any SETOSMC statements, OAM will not process second backup copies of objects.

You can use all of the keywords that are associated with the SETOSMC statement at both the global and the storage group levels. If you specify parameters without a corresponding storage group, OAM uses them as the default values for all Object storage groups in the configuration. If you specify parameters with a corresponding storage group, OAM designates those storage groups as the specific Object Backup storage groups for that Object storage group's backup copies.

See “SETOSMC Statements for Use in the OSMC Environment” on page 109 for a sample and syntax diagram of the SETOSMC statement. See “Displaying SETOPT, SETOAM, and SETOSMC Statements” on page 302 for descriptions of the SETOSMC statement parameters and command examples.

OAMXCF Statement for Parallel Sysplex Support

The CBROAM*xx* PARMLIB member contains one or more OAMXCF statements that provide support for a Parallel Sysplex environment. The OAMXCF statement allows you to provide group and member names for OAMplexes and instances of OAM to be associated with various XCF groups and members to allow data sharing within the sysplex. There are also timeout values that are assigned to optical and tape request types to determine the number of seconds that OAM is to wait for completion of a read or write request that was shipped to another OAM within the OAMplex. See “Using the UPDATE Command to Set OAMXCF Values” on page 314 for information on changing the OAMXCF values dynamically, or defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

ISMF Library Management Role within OAM

The storage administrator uses the Interactive Storage Management Facility (ISMF) Library Management to integrate OAM into system-managed storage. The storage administrator uses DB2 to define optical disk drives and libraries into the optical configuration database. The storage administrator then uses SMS to define the same drives and libraries into the specified SCDS, making such devices a part of the SMS configuration when that SCDS is activated.

In the case of object tape support, other information supplied by the SETOAM statement of the CBROAM*xx* PARMLIB member can supplement or override the ISMF information assigned to the Object or Object Backup storage group.

Note: OAM identifies tape volumes eligible for reading and writing objects through information provided by the Tape Volume Table in the optical configuration database and by the Object or Object Backup storage group to which the volume is assigned. Therefore, definition of tape libraries and tape drives to the optical configuration database through the use of ISMF is not required for object tape support. However, tape libraries are defined to an SCDS and to the tape configuration database in support of tape library management; this information is discussed in *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*. Tape drives are dynamically allocated by the system as needed to satisfy requests to read or write objects. For more information concerning how tape drives are allocated to the configuration, see “Dynamic Allocation of Tape Drives” on page 42.

Upon activation of an SCDS having optical libraries and optical disk drives defined, as well as tape drives available for allocation as needed, an operator on any console within an SMS complex can issue commands targeted for any library or drive within the configuration. For more information concerning ISMF usage in an optical environment, see “Appendix A. Sample Optical Hardware Configurations” on page 323. For further information regarding ISMF, refer to *z/OS DFSMS: Using the Interactive Storage Management Facility*.

Installation Storage Management Policy Overview

Each installation defines a storage management policy that allows effective object storage management without requiring user intervention. Through the use of ISMF, the storage administrator and system programmer define an installation storage management policy in an SMS configuration. OAM then manages object storage according to the currently active policy. Optical, tape, and DASD can all be used as the primary storage media for storing objects. Backup copies of objects can only be stored on optical or tape volumes. See Figure 3 on page 10 for a pictorial overview of this process.

An SMS configuration consists of the following elements:

- **Base configuration.** The base configuration identifies the systems in an SMS configuration and contains installation defaults. It also applies to SMS-managed data sets as well as objects. Only object-related functions are discussed in this document.
- **Automatic class selection (ACS) routines.** The storage administrator uses the ACS routines to assign storage group, storage class, management class, and data class constructs to data sets or objects that are based on customer-defined criteria. They are invoked with user-input variables, and they make decisions based on the environment called. The ACS routines use input values to set new values which causes changes in the SMS handling of the data.
- **Optical library and optical drive definitions.** Optical storage hardware must be defined to the system through ISMF before it can be used. (Remember, tape drives are dynamically allocated for use when required. They are defined to the system through the use of the Hardware Configuration Definition [HCD], not ISMF.)
- **Constructs.** Constructs are lists of attributes that are assigned to objects and storage areas. An SMS configuration can contain five types of constructs. The following describes all five constructs, including aggregate group; however, OAM uses only four of them (storage group, storage class, management class, and data class) in support of object storage management.
  - The **Storage group** construct allows you to define a storage hierarchy and manage that hierarchy as if it were one large, single storage area. See “Understanding the Storage Group Construct” on page 11 for information on establishing and manipulating storage groups.
  - The **Storage class** construct allows you to define different levels of performance objectives and availability requirements for objects. See “Understanding the Storage Class Construct” on page 16 for information on assigning an object to a storage class.
  - The **Management class** construct allows you to define backup, retention, and class transition attributes for objects. See “Understanding the Management Class Construct” on page 21 for specific information on defining the management class attributes.
  - The **Data class** construct allows you to define specific data attributes that are required for your installation’s tape storage. See “Understanding the Data Class Construct” on page 22.
  - **Aggregate group** allows you to group a collection of data objects that form a data type. This allows the data to be referred to collectively or individually. Aggregate group is *not used* within an OAM environment. It is used in conjunction with the storage of DASD data.

  An SMS configuration can contain multiple constructs of each type.

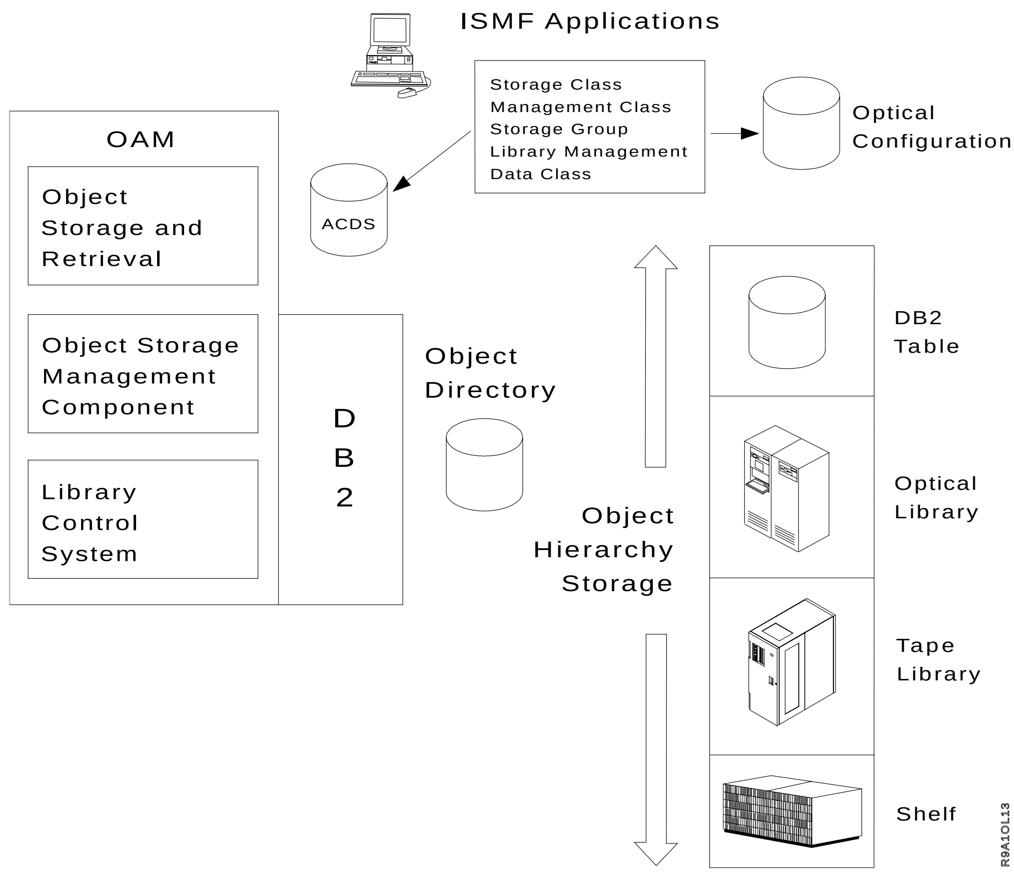

Figure 3. Overview of the Installation Storage Management Policy

This manual provides an OAM view of SMS. Refer to *z/OS DFSMS Introduction* for a more detailed introduction to SMS, the *z/OS DFSMSdfp Storage Administration Reference* for details on the topics introduced in this section, and the *z/OS DFSMS: Using the Interactive Storage Management Facility* for details on ISMF function.

SMS Constructs and ACS Routines

This section presents detailed information about the SMS constructs used by OAM and their relationship with the automatic class selection (ACS) routines. (See Figure 7 on page 24 for a diagram of this relationship.) OAM defines the management policy parameters in the SMS constructs of storage group, storage class, management class, and data class. The parameters include:

- Object retention rates
- The media on which OAM stores object collections
- Legal requirements for object retention
- Retrieval response time
- Location of object collections in the storage hierarchy
- How long OAM should hold the object collection at that level in the hierarchy
- Whether you need one or two backup copies of an object
- The media type to which OAM should direct backup copies of objects
- Affiliation of libraries with relevant storage groups

OAM associates these parameters with every object that it stores. The storage administrator defines the associations through ACS routines. The sections that follow describe these SMS constructs.

Understanding the Storage Group Construct

Before considering the specific interactions of storage groups and objects, it is worth reviewing basic storage group concepts as they apply to data sets and objects.

For data sets, the storage group construct helps simplify the task of administering external data. By putting a number of homogeneous data sets into one storage group, they can be viewed as one entity.

An Object storage group is composed of a set of volumes. Each installation develops Object storage groups according to its individual needs. Storage groups can be used to segregate different types of data (such as production versus development) and aggregate like types of data.

By separating the physical volumes from the service level (as defined by the storage class construct), Object storage groups also allow installations to change the physical aspects of storage without affecting the logical requirements of data access. For example, a new volume or device can be added to the storage group without affecting end-user routines.

In an OAM environment, Object storage groups allow the storage administrator to define an object storage hierarchy. The *object storage hierarchy* classifies storage areas according to location and, therefore, according to retrieval response time. Each object storage hierarchy must contain an *object directory*, containing control information about each object. Additionally, the hierarchy can have:
- DB2 object storage tables on DASD
- Optical volumes that are associated with optical libraries (real or pseudo), and stand-alone or operator-accessible optical disk drives
- Tape volumes that are associated with tape libraries or stand-alone tape drives

During an object's lifetime, it can move from one OAM storage hierarchy level (storage location) to another, ascending or descending depending on its performance objectives.

For more information on basic storage group concepts as they apply to data sets and objects, refer to *z/OS DFSMSdfp Storage Administration Reference*.

Object, Object Backup, and Tape Storage Groups

In addition to the storage groups that are defined by each installation for its data sets, OAM uses three special storage group types: Object, Object Backup, and Tape. OAM utilizes these storage groups as follows:
- An Object storage group is used to contain primary objects. See "Assigning Storage Groups" on page 12 for more information on assigning storage groups.
- An Object Backup storage group is used to contain the first or the second backup copy of each object for which the management class construct requires a backup. See "Assigning Object Backup Storage Groups" on page 13 for more information on assigning Object Backup storage groups.
- A Tape storage group contains tape volumes that are associated with an automated tape library dataserver (ATLDS) or a manual tape library (MTL). See "Assigning Storage Groups" on page 12 for more information on assigning storage groups.

A *primary* object is the primary copy of an object in the object storage hierarchy which is stored in the Object storage group on direct access storage device (DASD), optical, or tape. A *backup* object is the first backup copy of an object, which is stored in the Object Backup storage group specified as a first backup storage group. A *second backup* object is the second backup copy of an object, which is stored in the Object Backup storage group specified as a second backup storage group.

You can retrieve the primary, backup, or second backup copy of an object by using the OSREQ RETRIEVE command. Specify VIEW(PRIMARY), VIEW(BACKUP), or VIEW(BACKUP2) on the RETRIEVE request. See *z/OS DFSMS OAM Application Programmer's Reference* for more detailed information on the OSREQ macro. You can also get automatic access to the backup copies for retrieval by using the Automatic Access to Backup facility.

Defining an Object or Object Backup Storage Group

To define an Object or Object Backup storage group, use the ISMF Storage Group application. Use SETOAM statements in the CBROAM*xx* PARMLIB member to specify the tape-related options that can supplement or override these ISMF specifications for the Object or Object Backup storage group definitions. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for information on changing the SETOAM values dynamically, and on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

If a tape unit name is associated with an Object Backup storage group on the SETOAM statement in the CBROAM*xx* PARMLIB member, the backup copies are written to tape volumes. In this instance, any optical libraries that are associated with the Object Backup storage group that is defined using the ISMF storage group define panel are ignored for purposes of writing backup copies of objects to that Object Backup storage group. If the SETOAM statement does not direct the Object Backup storage group to tape media, then OAM writes the backup copies to optical media. Additionally, if there is no SETOAM statement in the CBROAM*xx* PARMLIB member, then OAM automatically writes the backup copies to optical media.

Other information that is supplied by the CBROAM*xx* PARMLIB member can supplement or override the ISMF information that is assigned to the Object or Object Backup storage group. See "Determining Which Media to Use for Backup Copies" on page 14 for more information on assigning backup copies of objects to various media types.

If an Object or Object Backup storage group belongs to an OAM that is a member of an OAMplex, it can be connected to more than one system in an SMS complex. The libraries that are defined for these storage groups can also be connected to multiple systems within the OAMplex. If the OAM is not part of an OAMplex, each Object or Object Backup storage group can be connected to only one system in the SMS complex.

Assigning Storage Groups

A Tape storage group can be assigned to an Multiple Virtual Storage (MVS) scratch tape when it is first used to store an OAM object. The Tape storage group is assigned to the tape volume through the ACS routines at tape volume allocation. A tape volume is assigned to an object backup storage group when space is needed to write a backup copy of an object. As a result, a single tape volume which was originally allocated inside of an ATLDS or MTL might have both an associated Tape storage group and an associated Object Backup storage group.

The Tape storage group is assigned to the tape volume, and the Object or Object Backup storage group is assigned to the object. Because a single tape volume can only contain objects that belong to one Object or Object Backup storage group, the tape volume is associated with the Object or Object Backup storage group containing the first object that is written to the tape volume.

An Object storage group is associated with an object through SETOSMC statements in the CBROAM*xx* member of PARMLIB. Through these statements, you may associate an Object storage group with a first and a second Object Backup storage group. If no storage groups are specified, then the defaults for the configuration are used.

Note: Processing is valid only for volumes that are associated with the Object or Object Backup storage groups that are defined in the active SMS configuration. A volume that is associated with an Object or Object Backup storage group that is not defined in the active SMS configuration cannot be accessed. Objects that are already written on that volume cannot be retrieved, and OAM cannot write new objects to that volume. A message (CBR0187I for optical volumes or CBR0210I for tape volumes) is issued at OAM initialization for each volume that is associated with the Object or Object Backup storage group that is not defined in the active configuration. To remedy this, define the Object or Object Backup storage groups to the active configuration. This is done through activation of an SCDS that contains the Object or Object Backup storage groups.

See “SETOSMC Statements for Use in the OSMC Environment” on page 109 for a sample SETOSMC statement that you can use when assigning storage groups to objects.

Assigning Object Backup Storage Groups

You can direct OAM to create a first and a second backup copy of objects using the existing number of backup versions field that is located in the SMS management class definition construct. OAM will now use this field to determine how many backup copies of an object are to be made when OSMC processing is done for an object’s storage group, if there are SETOSMC statements in the CBROAM*xx* member of PARMLIB indicating that second backup copies are to be created. See page “Defining Management Classes” on page 143 for the specifics of defining this field.

An Object Backup storage group can serve as either a first backup storage group, or as a second backup storage group, but not both. OAM will automatically verify that the Object Backup storage group is not specified as both a first and a second backup storage group.

Assigning Object Backup storage groups differs from assigning the Object storage group in that you must use SETOSMC statements to indicate that you want the second backup copy stored in an Object Backup storage group that is not the default Object Backup storage group. If you do not provide SETOSMC statements, then OAM will not process second backup copies of objects.

See “Displaying SETOPT, SETOAM, and SETOSMC Statements” on page 302 for the specifics of assigning Object Backup storage groups using SETOSMC statements.

Determining Which Media to Use for Backup Copies

OSMC uses the AUTO BACKUP parameter on the management class to determine if backup copies of an object should be written. OSMC will schedule writes of **two** backup copies if all of the following items are true:

- The AUTO BACKUP parameter equals **Y**.
- The number of backup versions that is specified in the management class field is greater than or equal to two (≥2).
- A SECONDBACKUPGROUP keyword is specified in a SETOSMC statement in the CBROAM*xx* PARMLIB member.

The backup copies of the object are written to the Object Backup storage groups assigned to the Object storage group to which the object belongs. Through the use of the SETOAM statements, you can specify that the backup copies of the object be written to the same removable media type or to different removable media types, or both. The media that is selected for the backup copies might be optical or tape. If OAM is initialized with a CBROAM*xx* PARMLIB member containing SETOAM statements for the Object Backup storage groups, and the SETOAM statements include valid TAPEUNITNAME specifications, the backup copies are written onto tape media. If there are not valid SETOAM statements for the Object Backup storage groups (this could also mean that the command is there, but the TAPEUNITNAME is not valid), all backup copies are written to optical media.

OSMC will schedule a write of only **one** backup copy if all of the following items are true:

- The AUTO BACKUP parameter equals **Y**.
- The number of backup versions specified is one.

OAM writes the single backup copy to the designated Object Backup storage group onto the media type that is assigned for that storage group.

Note: The default number of backup versions in the number of backup versions field is two.

First and second backup copies of objects can never reside on the same tape volume or optical disk volume because the first copy is stored on a volume belonging to the Object storage group and backup copies reside on volumes belonging to the specified Object Backup storage groups. A single tape volume or optical disk cannot belong to more than one Object or Object Backup storage group. Likewise, first and second backup copies of objects belong to separate object backup storage groups; therefore, first and second backup copies of objects cannot reside on the same volume.

The optical disk volume that is chosen to store the primary copy or first or second backup copies of the object will be an optical volume (primary copy) or backup optical volume (backup copy) that is contained in one of the real or pseudo optical disk libraries that are listed in the Object or Object Backup storage group definitions.

Allocating a Scratch Tape for the Tape Storage Group

In the case where a scratch tape volume is being allocated to store the primary or backup copy of the object, MVS scratch tape allocation chooses a library-resident tape volume (from an ATLDS or MTL) that is associated with the Tape storage group. If a scratch tape volume is not needed, an existing tape volume that is associated with an Object or Object Backup storage group that is defined in the current SCDS and capable of being mounted and handling the write request is allocated.

Using the DATACLASS Parameter to Determine Compaction

If the storage class indicates tape media (see "Media Selection for Object Storage" on page 16 for more information) and the Tape storage group that specifies the tape volume allocation is for a tape library dataserver, the allocation process uses the DATACLASS parameter on the SETOAM statement to determine tape compaction or no tape compaction for the volume. If the TAPECOMPACTION or NOTAPECOMPACTION keywords on the SETOAM statement are specified for a particular storage group, the data is written in compacted or noncompacted format as specified. If these keywords are not specified at storage group level of the SETOAM statement, the DATACLASS parameter of the SETOAM statement at the global level is used to determine tape compaction or no tape compaction. Should the DATACLASS parameter not be specified at the OAM global level, information that is passed on the DEVSUP parameter after MVS allocation processing is used to determine whether the allocated tape volume should have tape compaction or no tape compaction.

Using the TAPEUNITNAME Parameter for Volume Allocation to a Stand-Alone Drive

If the volume allocation is done using a stand-alone tape drive and no OAM scratch tape is available, OAM allocates a scratch tape using the TAPEUNITNAME parameter that is specified in the subparameter of the STORAGEGROUP parameter on the SETOAM statement. If no TAPEUNITNAME parameter is associated with the Object storage group that is assigned to the object, the storage of the object fails.

Note: The TAPEUNITNAME is stored in the UNITNAME column of the TAPEVOL table. The TAPEVOL table contains a single row for each tape volume containing OAM objects. The TAPEUNITNAME is required and is specified on all dynamic allocations so that the device which is allocated is compatible with the tape to be mounted.

Using SETOAM to Direct Objects to a Specific Device Type for an Object Storage Group

You can use a SETOAM statement to direct all objects for an Object or Object Backup storage group (which has already been directed to tape due to its associated storage class) to a specific device type by specifying the TAPEUNITNAME(*device-type*) parameter that is associated with the storage group.

The following are examples of directing objects to specific tape device types using the TAPEUNITNAME parameter of the SETOAM statement:

- If 3490 is the TAPEUNITNAME parameter, then objects written to the subject Object or Object Backup storage group would all have this unit name recorded in the tape volume table. Objects written to the Object or Object Backup storage group would all be written to 3490 devices.
- If the TAPEUNITNAME(3490) is changed to a new device type, such as TAPEUNITNAME(3590), OAM continues to write to the available usable tape volumes that are associated with the Object or Object Backup storage group using 3490 devices until an MVS scratch allocation is required. When an MVS scratch tape is required to handle an out-of-space condition for the Object or Object Backup storage group, that scratch tape is written on a device that is specified by the current SETOAM TAPEUNITNAME for that group, which is 3590 in this example.

  OAM continues to read the objects that were previously written on 3490 devices for the Object or Object Backup storage group so long as there is a 3490 device available for allocation at the time the read is requested. See “Dynamic Allocation of Tape Drives” on page 42 for information concerning compatible devices not being available to handle requests.

Note: If a tape volume that was previously written on a 3490 device is entered into an ATLDS or MTL, that tape volume continues to be used as long as the ATLDS or MTL to which it was entered contains a compatible device.

Grouping Devices (Esoteric Unit Names)

Devices can be grouped together and defined as one group to the system. For instance, a group of 3590 tape drives in the same room can be grouped together and defined as 3590GRP. These device groups are known as esoteric unit names or esoterics. Once an esoteric is specified on the SETOAM command for a group, it is necessary to ensure the existence of that esoteric as long as there is an OAM tape which has that esoteric specified in the UNITNAME field of the TAPEVOL table. It is also imperative that the contents of that esoteric not be changed to introduce incompatible device types. Should the esoteric name be deleted or changed, the volumes that are associated with the esoteric name cannot be allocated, because the TAPEUNITNAME cannot be resolved, so the tape volume that is required for the request is not mounted and the allocation request fails.

Using DB2 with Object Storage Groups

Each Object storage group consists of a DB2 database. The DB2 database is referred to as the *object storage database*. The object storage database contains the following DB2 tables:

- An object directory containing entries for objects residing in a particular Object storage group. These entries contain control information needed to locate and manage the object.
- A storage table for objects less than or equal to 3980 bytes.
- A storage table for objects greater than 3980 bytes.

Understanding the Storage Class Construct

The storage class construct enables storage administrators to separate the logical requirements for accessing data from the physical requirements for storing data. Storage class logically represents the level of service (performance objectives and availability requirements) for an object.

Every object is assigned to a storage class when it is created; therefore, every object is SMS-managed. This assignment determines where the object initially resides in the object storage hierarchy (optical, tape, and DASD). See Figure 7 on page 24 for a diagram of this process. The assignment can change as part of a class transition event or as the result of an explicit application request (OSREQ CHANGE). See “Modifying Default Storage and Management Classes” on page 167 for information on changing defaults if necessary.

For more information on using the OSREQ macro to customize your application interface, see *z/OS DFSMS OAM Application Programmer’s Reference*.

Media Selection for Object Storage

Each object that OAM stores is assigned a storage class and a management class. Storage class is used to determine the initial placement of an object in the OAM object storage hierarchy, and later is used during the OSMC storage management cycle to determine the correct current placement of the object when the storage management cycle processes that object. OAM uses the Initial Access Response (IARS) parameter in the storage class to determine if a primary copy of an object is stored on DASD or on removable media (optical and tape). If the IARS parameter in the storage class that is assigned to the object is zero, the primary copy of the object is stored in DB2 tables on DASD. If the IARS parameter is nonzero, the primary object is stored on removable media.

The Sustained Data Rate (SDR) parameter of the storage class is used to determine which removable media, optical or tape, is used to accept the primary copy of the object once it is determined that removable media is to be used. If the SDR parameter of the storage class is *greater than* or *equal to* three (≥3), the primary copy of the object is stored on a tape volume. If the SDR parameter of the storage class is *less than* three (<3), the primary copy of the object is stored on an optical disk volume.

Figure 4 on page 18 shows how OAM uses the IARS and the SDR parameters to determine where to store a primary object.

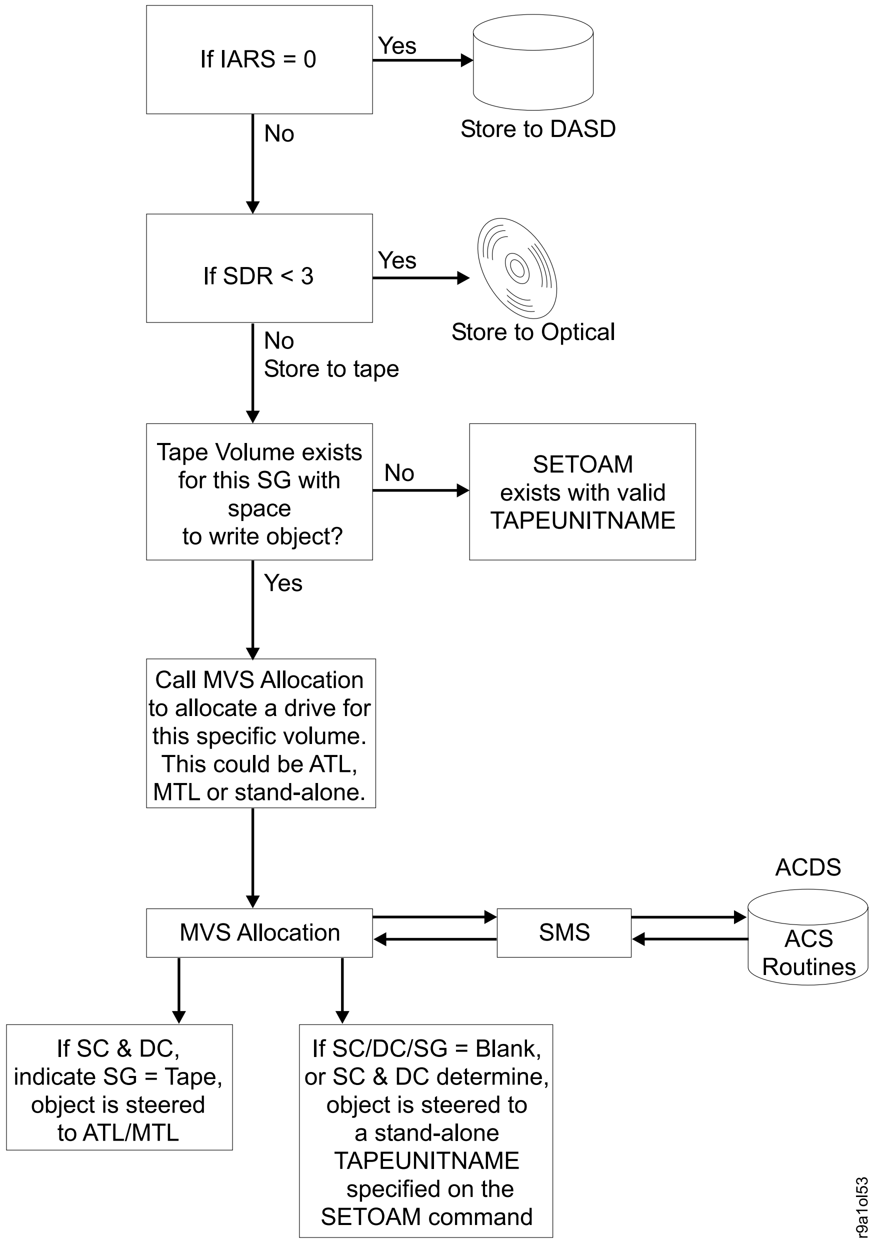

Figure 4. Conceptual Overview of Storing a Primary Object

Figure 5 on page 19 shows how OAM determines where to store a first backup copy of an object.

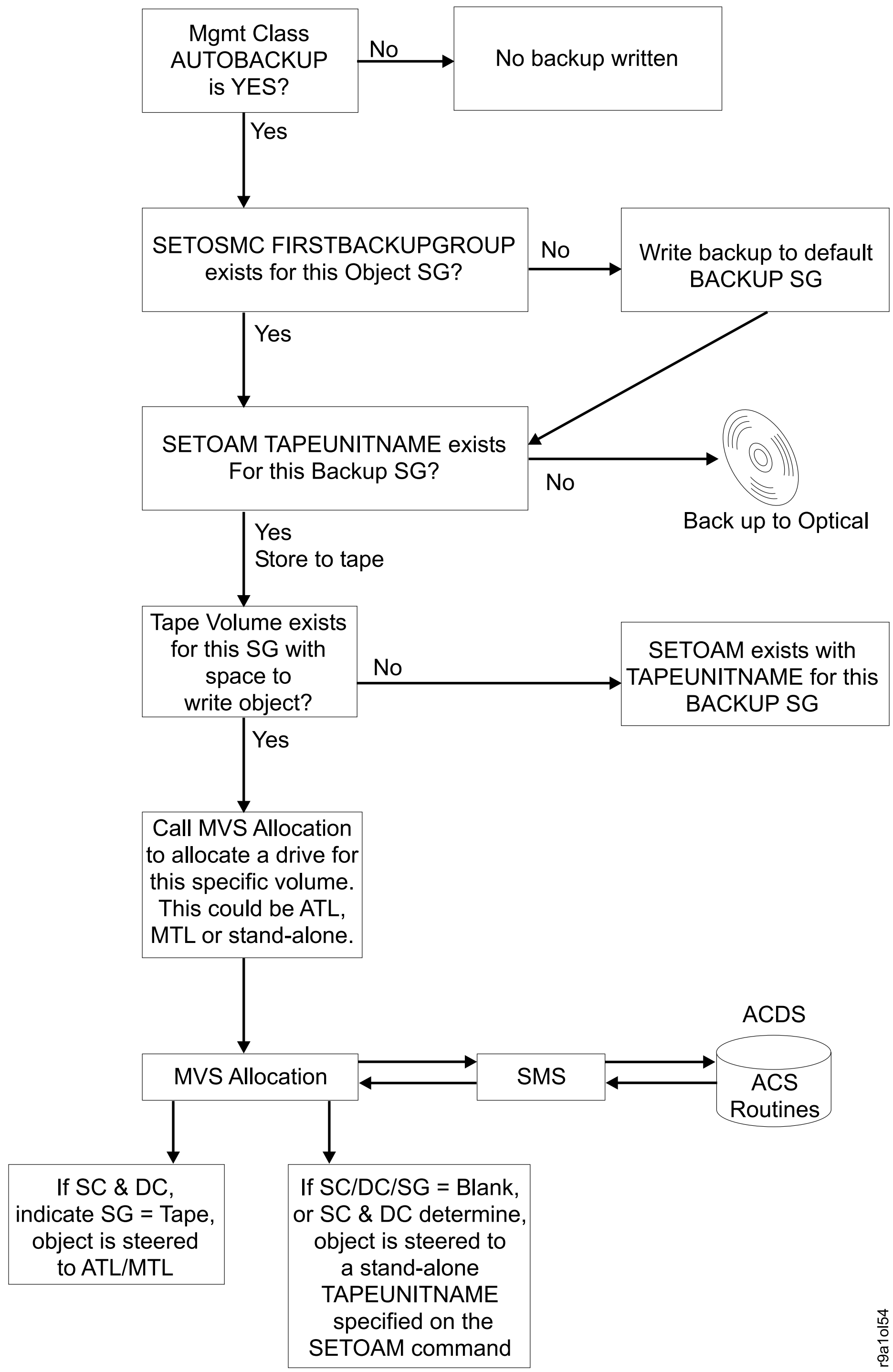

Figure 5. Conceptual Overview of Storing a First Backup Copy of an Object

Figure 6 on page 20 shows how OAM determines where to store a second backup copy of an object.

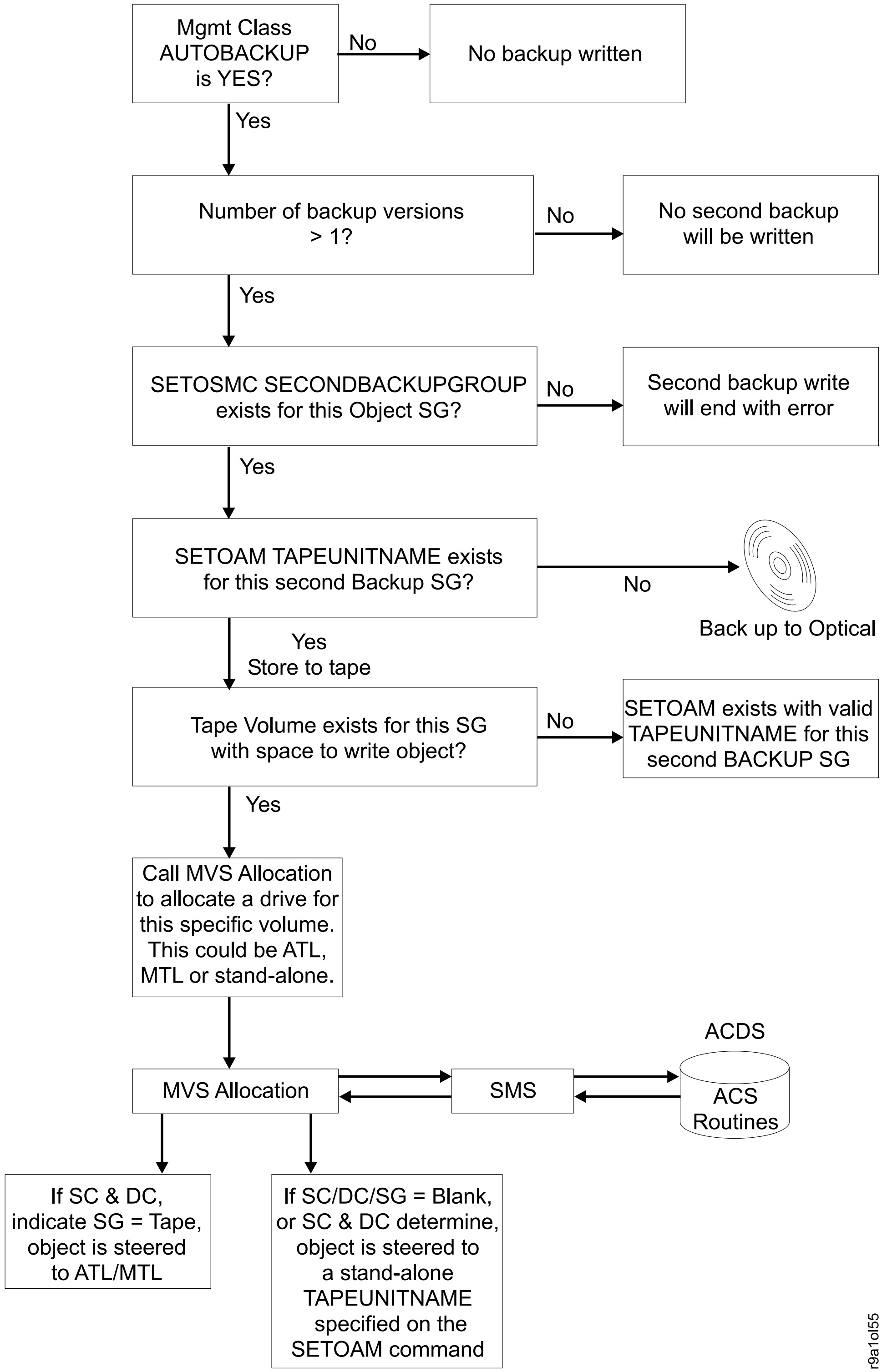

Figure 6. Conceptual Overview of Storing a Second Backup Copy of an Object

Object Storage on the Shelf

In addition to an object being stored inside a library (library-resident), an object can be assigned to a “shelf” (shelf-resident) location within the storage hierarchy. The following concepts apply to shelf storage:

- A shelf-resident volume can be either an optical or tape cartridge that is associated with stand-alone optical or tape drives or operator-accessible optical drives within a pseudo library. The optical volume is physically stored on a shelf location near the drives that are associated with the pseudo-library to which the volume is assigned. The tape volume is physically stored on a shelf location near the drives associated with the TAPEUNITNAME parameter to which the volume is assigned.
- Assigning an object to storage class of **SHELF** (through an OSREQ STORE/CHANGE request or a class transition) does not actually cause the object to be physically moved to another volume (the storage class ID row in the object directory table is the only change actually made). Nor does the volume, on which the object resides, automatically get ejected from the library to which it is associated, even if all the objects on the volume indicate a storage class of **SHELF**. There is nothing in the storage class definition that denotes shelf storage (no specific IARS value specific to shelf). This storage class assignment simply allows an installation a way of differentiating between performance objectives for objects that are actively accessed and those needing to be archived, or those that are accessed the least. It is up to the installation, in accordance with their storage management policy, to determine whether these objects having a storage class of shelf, should actually be removed from the library and placed physically on a shelf location for storage.

Understanding the Management Class Construct

The SMS management class construct is a list of class transition, backup, and retention attributes. OAM uses some management class attributes to manage objects. Every object is assigned a management class when it is created. See Table 2 on page 22 for management class examples, and “Modifying Default Storage and Management Classes” on page 167 for information on changing defaults if necessary.

Class transition attributes allow OAM to change the way an object is managed based on its age, its usage, or a predefined, periodic calendar event (for example, the first day of every month). Class transitions occur when the OSMC storage management cycle is invoked. This invocation is referred to as a “storage management cycle”. For objects requiring class transition, OSMC uses the ACS routines to determine if the objects should be managed using a different management class or if they should be placed at a different level of the storage hierarchy according to a different storage class.

OAM uses the AUTO BACKUP parameter and the NUMBER OF BACKUP VERSIONS parameter in the management class construct to decide whether to write one or two backup copies of an object. “Determining Which Media to Use for Backup Copies” on page 14 describes how OAM determines whether to write one or two backup copies. The backup copies can be made during the first storage management cycle after the object is stored, or during the first storage management cycle after a new management class is assigned for the object.

Expiration attributes of the management class definition determine the OAM action for object expiration. An object can expire automatically based on its age, its usage, or a specific date (derived from its management class or a management-class-approved object-specific retention period, if provided). OSMC deletes expired objects automatically during the storage management cycle, with the approval of an installation exit. For more information, see “Appendix E. Auto-Delete Installation Exit” on page 509.

Table 2. Management Class Examples for Objects

MANAGEMENT CLASS NAMES	TIME SINCE CREATION (retention attribute)	TIME SINCE LAST USE (transition attribute)	PERIODIC (transition attribute)	AUTOBACKUP (backup attribute)	NUMBER OF BACKUP VERSIONS (backup attribute)
MAGONLY	30 DAYS	N/A	N/A	NO	0
FOREVER	NOLIMIT	NOLIMIT	N/A	YES	2
MAGS	N/A	5 DAYS	N/A	NO	0
MAG30D	30 DAYS	N/A	N/A	YES	1
MAG30LIB	6 MONTHS	N/A	N/A	NO	0
MAG30SHF	7 YEARS	N/A	N/A	NO	0
OPT6LIBF	6 MONTHS	N/A	N/A	NO	0
OPT6SLH	7 YEARS	N/A	N/A	NO	0
TAPE30	30 DAYS	N/A	N/A	NO	0
TAPE30B	30 DAYS	N/A	N/A	YES	2
Note: N/A = Not Applicable.					

Understanding the Data Class Construct

Data class is an SMS construct that determines the characteristics for a tape volume during scratch tape allocation. Data class determines the following attributes for a tape volume allocated in an ATLDS or MTL:

- Tape expiration date
- Retention period
- Tape Device Selection Information (TDSI):
  - Compaction (YES | NO | BLANK)
  - Media type (BLANK | MEDIA1 | MEDIA2 | MEDIA3 | MEDIA4)
  - Recording technology (BLANK | 18-TRACK | 36-TRACK | 128-TRACK | 256-TRACK)
  - Special attribute (NONE | READCOMPATIBLE)

Data class determines the following attributes for a tape volume allocated to a stand-alone tape drive:

- Tape expiration date
- Retention period

Determining Data Class During Scratch Tape Allocation

The data class of a volume is determined when a scratch tape volume is allocated. If the allocation is steered to an ATLDS or a MTL, the data class subparameter on the SETOAM statement of the CBROAM*xx* PARMLIB member for the Object or Object Backup storage group is used if it is specified.

If the SETOAM statement does not specify the data class subparameter at the storage group level, the DATACLASS parameter of the SETOAM statement for the OAM global level is used to specify the values for the tape volume. The DATACLASS parameter of the SETOAM statement at the global level is applied to all tape volumes belonging to storage groups not having their own DATACLASS assigned. If there is no DATACLASS specification at the storage group or at the OAM global level, OAM relies on the DEVSUP parameter default passed to OAM during data set OPEN processing after allocation processing to determine tape compaction or no tape compaction for the tape volume. The ACS routines can also supplement or override the data class values that are specified for the tape volume from either the SETOAM statement or the DEVSUP parameter.

Note: Allowing the ACS routines to assign or change the data class assignment of an OAM tape volume is not recommended. An installation should ensure that its ACS routines do not change a DATACLASS specification for OAM object tape support data sets, including OAM.PRIMARY.DATA, OAM.BACKUP.DATA, or OAM.BACKUP2.DATA. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information either at the Object or Object Backup storage group level or at the OAM global level—whichever best suits the requirements for the tape volume that is being allocated. Allowing the ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specifies compaction). It is the installation's responsibility to ensure that its ACS routines are written to keep the data class construct for OAM tape volumes intact. See Figure 4 on page 18 for a diagram of the process of storing objects to tape.

Ignoring TDSI Data Class Information for a Stand-Alone Allocation

If the allocation is for a stand-alone tape drive, the information that is specified on the TAPEUNITNAME subparameter of the STORAGEGROUP parameter of the SETOAM statement is used to determine the specific device type that is allocated for the tape volume. Tape volumes that are allocated to stand-alone tape drives are not SMS-managed (the objects on the tape volumes are SMS-managed, but the tape volumes are not). Therefore, the TDSI information (media type, compaction, and track size) of the data class construct is not necessary for non-SMS-managed volumes. This information is ignored when an MVS scratch tape volume is allocated to a stand-alone tape drive, and only the tape expiration date and retention period information that are provided by the data class construct are used.

For more information concerning these data class attributes, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ACS Routines

Automatic class selection (ACS) routines determine, validate, or override the existing values that are assigned for the storage group, storage class, management class, and data class constructs for a collection. The classes assigned are those assigned to the collection or those that are explicitly stated on the OSREQ request to store or change an object. ACS routines are called to verify the storage class and management class that are stated on the OSREQ request. The ACS routines can accept the stated class, select a different one, or reject the stated class and return an error code.

Every object belongs to a collection. Each collection belongs to one and only one Object storage group. When an object is stored, it is automatically assigned to the Object storage group to which its specified collection belongs. Every object, when it is stored, is assigned a storage class and a management class. See Figure 7 on page 24 for a diagram of this process.

Note: Storage and management classes are optional on the OSREQ STORE request. If they are not specified, they are assigned the defaults from the collection. For a new collection, the ACS routines will be called to supply defaults for them.

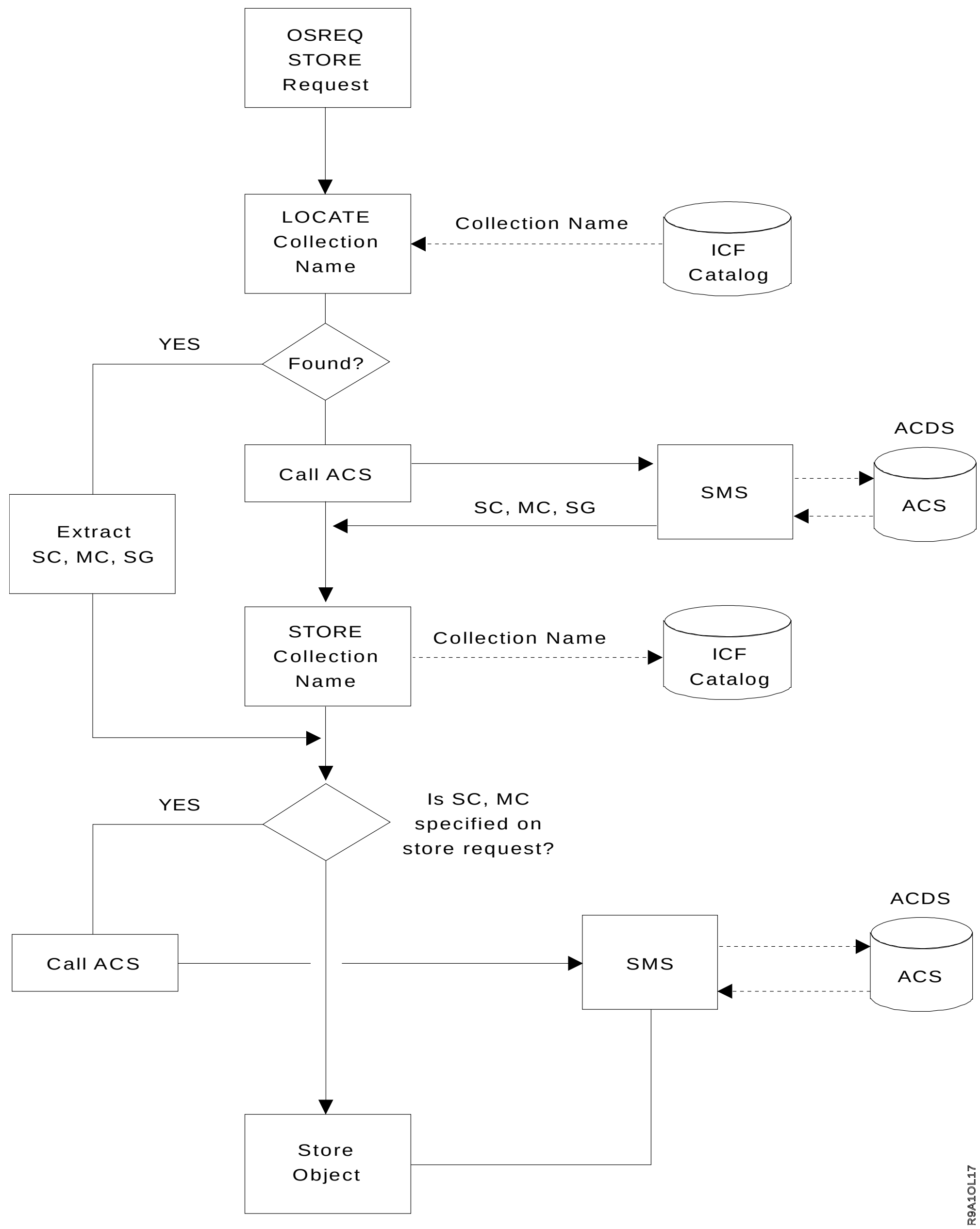

Figure 7. Storing an Object on Optical Media through an OSREQ STORE Request

A new tape data set name, OAM.BACKUP2.DATA, has been added for object tape backup volumes. To take advantage of the second backup copies of objects support, update ACS routines to accommodate this new data set name. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for details on this procedure.

In the case of objects residing on tape volumes, having the ACS routines alter the data class that was originally assigned to the tape volume through the SETOAM statement at MVS scratch tape allocation is not recommended. See the discussion regarding "data class and ACS routines" on page 23 for more information.

OAM Address Space

The OAM address space uniquely identifies the active OAM session. Start the OAM address space if you plan to use optical or tape devices for object support and OSMC functions or if you plan to delete objects within your data storage environment.

Note: There are four address space restrictions that you should consider for the duration of any single OAM session, regardless of whether it is part of an OAMplex:

- Only one OAM address space can be active in the MVS system.
- Only one DB2 subsystem can be associated with the active OAM address space.
- Optical devices can only be directly accessed by the owning OAM address space. However, requests for the optical devices can be shipped to the OAM in an OAMplex that owns the optical devices by using the XCF messaging service.
- Specifying a region size other than 0MB (zero megabytes) on the OAM started procedure JCL may result in storage shortage abends, especially during an OSMC cycle.

OTIS Address Space

The OAM thread isolation support (OTIS) is used when new collections are added to the catalog. OTIS provides an interface to DB2 to allow the collection table in DB2 to be updated. The OTIS address space uniquely identifies the active OTIS interface session. Start the OTIS address space if you plan to use any of the functions that are associated with the OSREQ application interface. See *z/OS DFSMS OAM Application Programmer's Reference* for further information on the OSREQ application interface.

Optical Storage

Optical disks are generally used for storing objects that are accessed infrequently, primarily because of their high capacity and performance characteristics. This section provides an overview of optical storage and its role in OAM. The optical devices that are supported by OAM are listed in Table 3.

Table 3. Optical Devices Supported by OAM

Device Name	Disk Size	Storage Slots	Internal Drives	Operator-Accessible	Comments
3995-131	5.25 inch	144	4	1	Attaches to host system. Supports sd-REWR media.
3995-132	5.25 inch	144	4	1	Attaches to host system. Supports sd-WORM media.
3995-133	5.25 inch	144	4	1	Attaches to host system. Supports sd/dd WORM/REWR media.
3995-111	5.25 inch	144	4	0	Attaches to Model 131. Supports sd-REWR media.
3995-112	5.25 inch	144	4	0	Attaches to Model 132. Supports sd-WORM media.
3995-113	5.25 inch	144	4	0	Attaches to Model 133. Support sd/dd WORM/REWR media.

Table 3. Optical Devices Supported by OAM (continued)

Device Name	Disk Size	Storage Slots	Internal Drives	Operator-Accessible	Comments
3995-C3A	5.25 inch	0	0	1 or 6	Attaches to host system. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C32	5.25 inch	52	2	0	Attaches to Model C3A. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C12	5.25 inch	52	2	0	Attaches to Model C3A/C32 pair. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C34	5.25 inch	104	2 or 4	0	Attaches to Model C3A. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C36	5.25 inch	156	4 or 6	0	Attaches to Model C3A. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C16	5.25 inch	156	4 or 6	0	Attaches to Model C3A/C36 pair. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C38	5.25 inch	258	4 or 6	0	Attaches to Model C3A. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-C18	5.25 inch	258	4 or 6	0	Attaches to Model C3A/C38 pair. Supports sd*/dd/qd/8x** WORM/REWR media.
3995-SW3	5.25 inch	N/A	N/A	N/A	Supports sd*/dd/qd WORM/REWR media. This is the drive used in the C3A, C1x, and C3x libraries.
3995-SW4	5.25 inch	N/A	N/A	N/A	Supports sd**/dd/qd/8x WORM/REWR media. This is the drive used in the C3A, C1x, and C3x libraries.
9246	12 inch	64	2 or 4	0	Attaches to host system. Supports WORM media.
9247	12 inch	0	0	N/A	This is the drive used in the 9246 library. Supports WORM media.

Table 3. Optical Devices Supported by OAM (continued)

Device Name	Disk Size	Storage Slots	Internal Drives	Operator-Accessible	Comments
Note: MB = 1 048 576 bytes GB = 1 073 741 824 bytes REWR = rewritable WORM = write-once, read-many sd = single-density (652 MB maximum disk size) dd = double-density (1304 MB maximum disk size) qd = quad-density (2600 MB maximum disk size) 8x = eight times density (5.2 GB maximum disk size) For information concerning approximate user data disk capacities for the supported media, see Table 17 on page 75. The approximate user data disk capacity can vary depending on file sizes. Smaller file sizes take up more space on the disk than larger file sizes due to the increase in extents. Refer to *3995 Introduction and Planning Guide* or *3995 Introduction and Planning Guide for C-Series Models* for more information on this topic. * The 3995–SW3 drive (the drive in the C3A, C1x, and C3x libraries) is only capable of reading from single-density WORM and rewritable media, not writing to it. It is capable of reading from or writing to double- and quad-density media. **The 3995–SW4 drive (the drive in the C3A, C1x, and C3x libraries) is only capable of reading from single- and double-density WORM or rewritable media, not writing to it. It is capable of reading from or writing to quad- and 8x-density media. Quad-density and 8x-density drives cannot coexist in the same library.					

Note: For simplicity, the following 3995 library models are referred to as C1x and C3x respectively within this publication: C12, C16, C18, C32, C34, C36, C38.

Optical Disk Cartridges

Objects are stored on optical media called *optical disks*. Each optical disk is encased in a protective housing. Together, the disk and its housing are called an *optical disk cartridge*. An optical disk has recording surfaces on both sides. Each side is referred to as an *optical volume*. For information concerning optical disk cartridge capacities, see Table 3 on page 25, or refer to *3995 Introduction and Planning Guide*, or *3995 Introduction and Planning Guide for C-Series Models.*

Optical Recording Techniques

Using laser technology, optical disk cartridges access *optical disk drives* to seek, read, write, and delete data on optical disks through the means of two optical media recording processes:

- Write-Once, Read-Many (WORM) technology
  - Reads from and writes to 5.25-inch, 12-inch, single-, double-, quad-, and 8x-density WORM optical disk media.
- Magneto-Optic (MO) rewritable technology
  - Reads from and writes to 5.25-inch, single-, double-, quad-, and 8x-density, rewritable optical disk media.

    Notes:

    1. The term “rewritable” is used within this manual to depict this type of optical disk media. Also, continuous composite WORM (CCW) media is included wherever the terminology double-, quad-, and 8x-density WORM media is used, unless otherwise stated.

2. The 3995-SW3 optical disk drive (in all the C3A, C1x, and C3x libraries) is not capable of writing to any 3995 single-density (WORM or rewritable) media. It is capable of reading this type of media, as well as reading from and writing to 3995 double- or quad-density WORM, rewritable, or CCW optical disk media type.
3. The 3995-SW4 optical disk drive (in all the C3A, C1x, and C3x libraries) is not capable of writing to any 3995 single-, or double-density (WORM or rewritable) media. It is capable of reading this type of media, as well as reading from and writing to any 3995 quad- or 8x-density WORM, rewritable, or CCW optical disk media type.

Write-Once, Read-Many Recording Technique

Write-once recording is an irreversible process that uses heat from a laser beam to make holes in the surface of the optical disk. Once the record is created, it cannot be altered. If the data needs to be written again, a new record is created, but the space used by the original entry is not recovered. This type of media is advantageous in instances where a permanent record is needed (for example, signed application forms), or when data is stored that will never be altered or updated (for example, in the case of items being stored on microfiche, completed forms, or X-rays). Because of the permanent nature of the data recorded, you can access WORM optical disks an unlimited number of times (read-many). See Figure 8 for a graphical depiction of WORM technology.

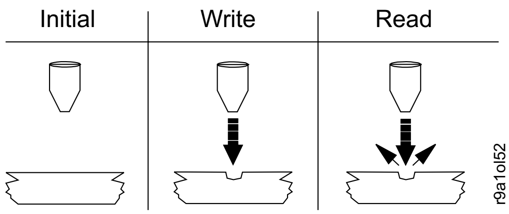

Figure 8. Write-Once, Read-Many (WORM) Recording Technique

Magneto-Optic Rewritable Recording Technique

Magneto-Optic (MO) rewritable recording is a reversible process that combines the use of magnetic and laser technologies to write, read, erase, and rewrite data. Rewritable recording is somewhat similar in concept to DASD recording. It uses a laser beam to heat the recording layer and then applies a magnetic field. The direction of magnetization changes only when the media is heated and the magnetic field is applied simultaneously. This process is used both at the time of recording and at the time of erasure. See Figure 9 for a pictorial overview of the MO optical media recording process.

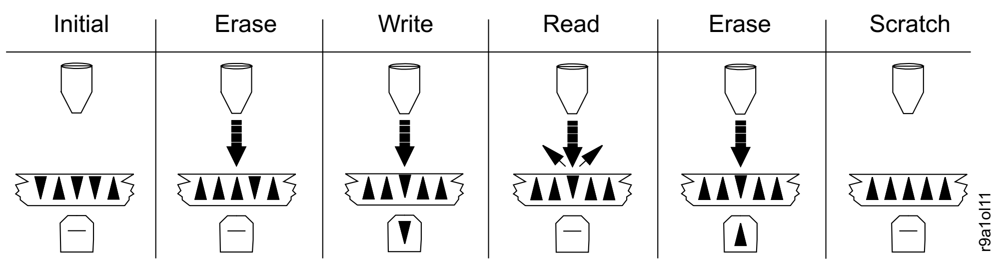

Figure 9. Rewritable Magneto-Optic (MO) Recording Technique

Optical Volumes

Each optical volume is identified by a unique volume serial number. This unique volume serial number must be unique within the entire installation. This means that no two volumes (regardless of the media type) can have the same volume serial number. The volume serial number must conform to MVS volume serial number naming conventions as documented in the *z/OS MVS JCL Reference*.

Because optical volumes can reside either inside libraries or on shelves, the physical procedure for accessing volumes varies according to their location:

- When a library-resident optical volume is needed, the system mounts it on an optical disk drive in the optical library containing the volume. If a compatible optical library drive is unavailable (for example, the drive is offline or non-operational), the request to retrieve the object fails.
- When a shelf-resident optical volume is needed, the system requests that the volume be mounted on a stand-alone optical disk drive for a 9246 volume or on an operator-accessible optical disk drive for a 3995 volume. If a stand-alone or operator-accessible optical disk drive is unavailable (for example, the drive is offline or non-operational), the request to retrieve the object fails. If the requested volume is unavailable, the operator can terminate the request.

  Note: Shelf storage can be local to the computer facility and, therefore, accessible to the optical disk drive operator, or it can be located elsewhere.

The operator can enter a shelf-resident volume into any compatible optical library by placing the optical disk cartridge into the input/output station of the optical library. However, if there is an outstanding request to mount the volume on a stand-alone or an operator-accessible optical disk drive, the attempt to insert the volume into the library is rejected.

Optical Volume Types

There are three types of optical volumes:

- Scratch
- Grouped
- Backup

Note: Both volumes comprising an optical disk *must* belong to the same Object or Object Backup storage group.

Scratch Volumes: A scratch volume is an optical volume that is not yet associated with an Object storage group or an Object Backup storage group.

If an optical volume is not pre-formatted and labeled, it must be labeled with a unique volume serial number before it can be accessed by OAM. This task can be completed in either of two ways:

- Labeling an unlabeled optical disk on either a stand-alone or an operator-accessible optical disk drive in response to a LABEL command which is entered by the operator
- Labeling an unlabeled optical disk on a library-resident optical disk drive as a result of being inserted into the input/output station of an optical library

In either case, the operator is asked to supply the volume serial number for each side of the optical disk. The volume labels are written and rows are created in the volume table in the optical configuration database. The volumes are marked as scratch volumes or grouped volumes depending on how the operator replies to the CBR4432D message during label processing.

Both sides of a scratch volume become either grouped or backup volumes when a scratch volume is used to satisfy an out-of-space condition in an Object or an Object Backup storage group.

Note: Because WORM optical volumes that are full or have very little free space are not as useful as scratch volumes, the operator is notified by message CBR4451I if the kilobytes that are free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. This gives the operator the opportunity to fail the cartridge entry process by responding through message CBR4452D, thus causing the cartridge to be ejected from the library.

Grouped Volumes: A grouped volume is an optical volume that is associated with an Object storage group. Volumes are grouped to subdivide the total available optical storage. A grouped volume contains objects from a single Object storage group.

A scratch volume becomes a grouped volume when OAM uses it to satisfy a write request that specifies an Object storage group name. When a scratch volume becomes a grouped volume, both volumes on the optical disk become grouped volumes that are assigned to the same Object storage group.

Taken together, all of the optical volumes that are associated with an Object storage group constitute the optical volume portion of the hierarchy in that Object storage group.

Backup Volumes: A backup volume is an optical volume that is associated with an Object Backup storage group. Backup volumes are usually intended to provide disaster recovery or to meet legal storage requirements. They contain backup copies of objects whose primary copies reside elsewhere in the object storage hierarchy. OAM can create up to two backup copies of an object. A primary copy of an object resides on a grouped volume; backup copies reside on backup volumes.

A scratch volume becomes a backup volume when it is used by OAM to satisfy a write request for the Object Backup storage group. When a scratch volume becomes a backup volume, both volumes on the optical disk become backup volumes.

Optical Media Types

There are a number of optical disk media types that can be used with the 9246/9247 and 3995 optical devices. The following media types are all supported in the optical environment:
- 12-inch, WORM optical disk media
- 5.25-inch, single-density, WORM optical disk media
- 5.25-inch, double-density, WORM optical disk media
- 5.25-inch, single-density, rewritable optical disk media
- 5.25-inch, double-density, rewritable optical disk media
- 5.25-inch, quad-density, WORM optical disk media
- 5.25-inch, quad-density, rewritable optical disk media
- 5.25-inch, 8x-density, rewritable optical disk media
- 5.25-inch, 8x-density, WORM optical disk media

Note: Unless otherwise stated, continuous composite WORM (CCW) media is included wherever the terminology double-, quad-, and 8x-density WORM media is used. For more information on these media types and the libraries and drives that support them, see Table 3 on page 25.

The 9246 optical library and 9247 optical drive support the 12-inch WORM optical disk media.

The 3995 optical library dataservers support the following 5.25-inch optical disk media types:
- 3995 Models 132 and 112 support single-density, WORM optical disk media.
- 3995 Models 131 and 111 support single-density, rewritable optical disk media.
- 3995 Models 133 and 113 support single-density, double-density, WORM, and rewritable media.
- The 3995 Models C3A, C1x, C3x library models with the 3995-SW3 optical disk drive (which can be either an internal optical disk drive for the C3A, C1x, or C3x library models or an operator-accessible optical disk drive) support single-density read-only media, and double- and quad-density WORM and rewritable media.
- The 3995 Models C3A, C1x, C3x library models with the 3995-SW4 optical disk drive (which can be either an internal optical disk drive for the C3A, C1x, or C3x library models or an operator-accessible optical disk drive) support single- and double-density read-only media, and quad- and 8x-density WORM and rewritable media.

When an object is stored using the OSREQ STORE macro, it is assigned to a specific Object storage group by the SMS storage group ACS routine. If the object is stored on an optical volume, OAM selects an optical volume residing in one of the optical libraries associated with the Object storage group to which the object has been assigned. See page 356 for more information concerning the DEFAULT MEDIA TYPE option.

Optical Disk Drives

An optical disk drive uses laser technology to write data to and read data from an optical disk. The optical disk drives supported by OAM use removable media. The following are the different types of optical disk drives:
- Library-resident (in 3995 and 9246 libraries)
- Operator-accessible (3995)
- Stand-alone (9247)
- Multifunction (library resident or operator-accessible in all 3995 models except 3995-131 and 3995-132)

Library-Resident Optical Disk Drive

A library-resident optical disk drive is inside an optical library. The cartridge transport mechanism in the library mounts and demounts the optical disk cartridges for the internal disk drives.

Operator-Accessible Optical Disk Drive

In addition to internal disk drives accessing the optical disk cartridges stored inside the library (library-resident), an operator-accessible optical disk drive is provided for users who need to read, write, or delete from an optical disk without storing or retrieving it from the library. When WORM optical media is used, the data is logically deleted from the optical disks. When rewritable optical media is used, the data is physically as well as logically deleted from the optical disks.

The operator-accessible drive cannot be accessed by the library's internal cartridge transport mechanism, and the operator-accessible drive does not have access to the optical disk volumes which are stored inside of the library. Operator-accessible drives are used to support shelf-resident optical disks. A human operator mounts and demounts the shelf-resident optical disk cartridges for the operator-accessible optical disk drives.

Stand-Alone Optical Disk Drive

A stand-alone optical disk drive resides in a separate unit outside of an optical library. An operator mounts and demounts shelf-resident, optical disk cartridges for stand-alone optical disk drives. The 9247 it is an example of a stand-alone optical disk drive.

Multifunction Optical Disk Drives

The optical disk drives 3995-133, -113, -SW3, and -SW4 are considered *multifunction* drives. These drives are capable of being library-resident optical disk drives, or operator-accessible optical disk drives, or both. Multifunction drives are capable of reading and writing a combination of media types.

The 3995-133 and -113 optical disk drives support the following media types:
- Single-density, WORM, 5.25-inch optical disk media
- Single-density, rewritable, 5.25-inch optical disk media
- Double-density, WORM, 5.25-inch optical disk media
- Double-density, rewritable, 5.25-inch optical disk media

The 3995-SW3 optical disk drive supports the following media types:
- Single-density, WORM, 5.25-inch optical disk media (read-only)
- Single-density, rewritable, 5.25-inch optical disk media (read-only)
- Double-density, WORM, 5.25-inch optical disk media
- Double-density, rewritable, 5.25-inch optical disk media
- Quad-density, WORM, 5.25-inch optical disk media
- Quad-density, rewritable, 5.25-inch optical disk media

The 3995-SW4 optical disk drive supports the following media types:
- Single-density, WORM, 5.25-inch optical disk media (read-only)
- Single-density, rewritable, 5.25-inch optical disk media (read-only)
- Double-density, WORM, 5.25-inch optical disk media (read-only)
- Double-density, rewritable, 5.25-inch optical disk media (read-only)
- Quad-density, WORM, 5.25-inch optical disk media
- Quad-density, rewritable, 5.25-inch optical disk media
- 8x-density, WORM, 5.25-inch optical disk media
- 8x-density, rewritable, 5.25-inch optical disk media

The multifunction optical disk drive capability provides the flexibility to populate the libraries containing these optical disk drives with any combination of valid optical disk media. This can be done by using the DEFAULT MEDIA TYPE. See page 356 for information concerning DEFAULT MEDIA TYPE, and “Defining Optical Drives” on page 363 for information on defining multifunction optical disk drives.

| Due to the flexibility of the multifunctional optical disk drive, it is possible to load different optical disk media types into a single optical library dataserver. The user can then direct data for one application to WORM optical disk media, while routing data for another application to rewritable optical disk media within the same optical library dataserver. This can be done by performing both of the following actions:

| 1. Set up Object storage groups to direct the data to the appropriate media type.
| 2. Reply to message CBR4432D with the name of the Object or Object Backup storage group to which the volumes are to belong. OAM displays message CBR4432D if you are entering unlabeled optical disks into an optical library:

```
CBR4432D  Enter storage group name for volumes volser-1 and volser-2,
or reply 'U' to assign to scratch.
```

Optical Libraries

Specifically, an optical library is a device that houses one or more optical disk drives and cartridges, an optical disk cartridge storage area, an input/output station for inserting and removing optical disk cartridges, and a mechanism for moving optical disks between those areas.

In general, an optical library is a set of optical volumes and the optical disk drives that are associated with those volumes. The volumes within the optical library are said to be library-resident optical volumes. Optical volumes can also be located outside of the optical library. These volumes are referred to as shelf-resident optical volumes.

Shelf-resident optical volumes can be associated with stand-alone or operator-accessible optical disk drives, or both, that are used to create a pseudo optical library (see “Pseudo Optical Library Concept” on page 34 for more information).

An optical library can contain optical volumes belonging to more than one Object storage group or the Object Backup storage group, or both.

A 3995 optical library can be specified as connected to more than one system within an OAMplex. However, the library must still be *physically* connected to only one MVS/ESA system in an SMS complex at a time.

Optical libraries are defined to SMS and OAM using the ISMF Library Management application. See “Defining 9246 Optical Libraries” on page 328 and “Sample ISMF Session for an IBM 3995 Optical Library Dataserver” on page 351 for more information.

Notes:

1. To avoid confusion, do not assign the same name to libraries and Object storage groups when defining them to the optical configuration database.
2. Library names must be included in the Object and Object Backup storage group definition in order for optical writes to be performed for the Object or Object Backup storage group.

Real Optical Libraries

A real optical library (see Figure 10 on page 34) is a storage device containing the following elements:

- An input/output station for entering into and removing cartridges from the library
- Optical disk drives
- A cartridge storage area for holding optical disk cartridges
- A cartridge transport mechanism for moving cartridges between the input/output station, slots in the cartridge storage area, and the optical disk drives

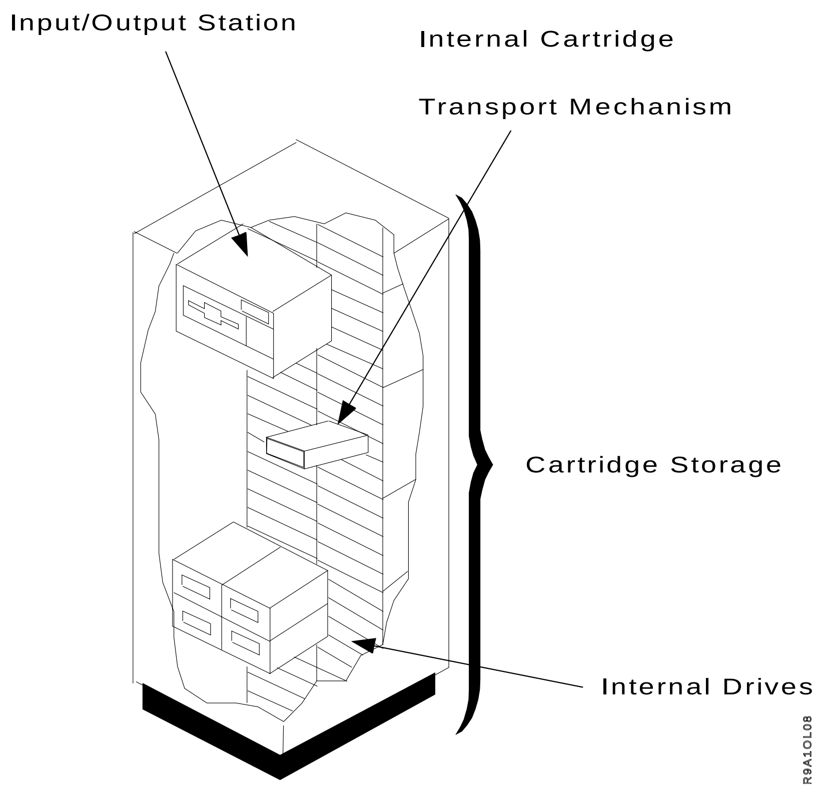

Figure 10. Real Optical Library

In real optical libraries, there is an implied affinity between the volumes and drives in the same library. In other words, an optical volume in an optical library can be mounted only on an optical disk drive within the same optical library.

Note: If there is a grouped request, OAM selects a volume that is in a library spanned by the Object or Object Backup storage group targeted for a write request. If a volume is in a library that is not included in the ISMF definition for the targeted Object or Object Backup storage group, the volume is not considered for write requests associated with that Object or Object Backup storage group. Any objects already written on the volume can still be read.

Pseudo Optical Library Concept

In addition to real optical libraries, OAM also supports the concept of pseudo optical libraries. The concept of pseudo libraries has changed from previous releases; however, pseudo libraries defined in previous releases are still supported.

A pseudo library is a collection of stand-alone or operator-accessible drives, or both, and shelf-resident volumes, defined by the installation and not necessarily of the same device and media types. Pseudo libraries are defined without a device type. When operator-accessible drives are defined, they can be assigned to a pseudo library chosen by the person defining the devices. These devices can be grouped in a manner that best fits the needs of the installation (for example, physical location, device and media affinity, backup and primary objects stored together, and so forth). There is no limit on the number of pseudo libraries that can be defined within an active SMS configuration.

Pseudo libraries defined prior to DFSMS 1.5.0 are still supported. These pseudo libraries were defined as a collection of one or more stand-alone or operator-accessible drives of the same device type, and one or more shelf-resident optical volumes of a like media type. OAM continues to honor these old pseudo library definitions. Support for both concepts allows installations wishing to convert their environments to the new pseudo library concept over a period of time the ability to use their previously defined pseudo libraries during the transition period.

It is the responsibility of the installation to determine the pseudo library to which an optical volume is to be associated. This determination is made either when the volume is ejected from a real library (see “Associating Pseudo Libraries” on page 254), or when a volume is labeled on an operator-accessible drive (see “Labeling an Optical Disk on a 3995 Operator-Accessible Drive” on page 259).

Note: It is recommended that you convert to the new pseudo library concept to allow more flexibility and customization within your storage environment. This conversion will prepare you for the eventual elimination of the device type association restrictions required with previous pseudo libraries.

During OAM initialization, if a volume record is encountered with an associated library name that is not known to the current configuration (ACDS), a message is issued indicating that the volume is ignored and that the library must be defined to the configuration before the volume will be recognized. For a shelf-resident volume, the library name associated with a volume is that of the pseudo library.

The default pseudo library definitions created by OAM are temporary definitions that exist only while the OAM address space is active. They are not part of the active SMS configuration (ACDS) or part of the SMS SCDS that was most recently activated. If no pseudo library is defined within the optical configuration database, OAM defines a pseudo library for each supported drive type. The reserved optical library names for these OAM-defined pseudo libraries for the shelf-resident volumes are as follows:

- STDALONE for 9247 drives and volumes
- PCTREUSE for 3995-131 drives and single-density rewritable volumes
- PCTWORM for 3995-132 drives and single-density WORM volumes
- P3995133 for 3995-133 drives and double-density volumes
- P3995SW3 for 3995-SW3 drives and quad-density volumes
- P3995SW4 for 3995-SW4 drives and 8x-density volumes

Attention: It is the responsibility of the installation to make certain that there are drives capable of handling the media types within the pseudo library so that OAM can direct requests to a compatible drive for the task. Otherwise, the request fails, and an error message indicates that there were no available drives for the request.

Associating Ejected Optical Volumes with Pseudo Libraries

Once an optical volume is ejected from a real optical library, it becomes shelf-resident and is associated with a pseudo optical library determined by the installation. For more information concerning associating ejected volumes with pseudo libraries, see “Associating Pseudo Libraries” on page 254.

Mounting a Shelf-Resident Optical Volume on an Operator-Accessible Drive

When a read request for a 3995 shelf-resident volume that belongs to a default pseudo library (with device type association) is received by OAM, any operator-accessible drive belonging to a pseudo library with an associated device type that is read-compatible with the volume is eligible to mount the volume.

When a read request for a 3995 shelf-resident volume that belongs to a pseudo library with no device type association is received, any operator-accessible drive (within the set of drives assigned to that pseudo library) with a device type that is read-compatible with the volume is eligible for the request.

If the request is a write request, drive selection is based on the drives in libraries associated with the storage group. A drive must belong to a library that is associated with the storage group in order to be considered. If a volume belonging to a pseudo library with a device type association is selected, only drives that are write-compatible and belong to a pseudo library that also has a device type association are eligible for the request. If the volume selected belongs to a pseudo library that has no device type association, then only drives that are write compatible with the selected volume and belong to the same pseudo library as the volume are eligible for the request. This makes it possible to isolate volumes and operator-accessible drives by physical location instead of by device type. An installation can choose to have a large pseudo library if everything is in the same location, or have several pseudo libraries in various locations as long as there are associated drives that can satisfy the request.

When a shelf-resident optical volume is mounted on an operator-accessible drive in an OAMplex environment, the volume is managed and controlled by the instance of OAM to which the drive belongs. Any requests for the volume are then sent to the OAM where the volume is currently mounted, thus eliminating the need to demount and remount the volume.

Failing Read/Write Requests for Pseudo Libraries

If a shelf-resident optical volume is associated with a pseudo optical library that does not contain any stand-alone or operator-accessible optical disk drives, then requests to write data to that volume or requests to read data from that volume fail because there is no optical disk drive on which to mount the volume. OAM never asks for a specific shelf-resident optical disk volume (by volume serial number) to be entered into a specific optical disk library for the purposes of reading data from or writing data to the volume.

Tape Storage

Tape volumes provide a low cost storage medium for storing primary and or backup copies of objects. Storing objects on tape volumes in conjunction with DASD and optical media provides flexibility and efficiency within the storage management facility. All devices within the installation can be used in concert with each other, complementing the performance objectives of the objects that reside on each medium. Objects can be migrated from DASD to tape to optical disk or any combination of these three media, providing the most cost-effective method for meeting your data storage objectives. The following information provides an overview of tape storage and role it plays in object tape support.

Table 4 on page 37 is referenced throughout this publication and can be useful in planning for tape storage within your environment. The table provides an overall detailed reference into the tape devices, media, and pertinent information associated with tape storage supported by OAM.

The hardware configurations described in Table 4 on page 37 can be used separately or in specific combinations to create or modify your tape storage environment.

Table 4. Tape Storage Configurations

Library Model	Subsystem Device Type	Library Attachment	Media Supported		Recording Technology	Noncompacted Data Capacity
3495 L20, L30, L40, L50	3490	Yes	MEDIA1	(R/W)	18	200MB
	3490E	Yes	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
	3590-1*	Yes	MEDIA3/4	(R/W)	128	10GB, 20GB
3494 L10	3490E	Yes	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
	3590-1*	Yes	MEDIA3/4	(R/W)	128	10GB, 20GB
	3590-E** (3590-1 emulation)	Yes	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10GB, 20GB 20GB, 40GB
MTL	3480	N/A	MEDIA1	(R/W)	18	200MB
	3490	N/A	MEDIA1	(R/W)	18	200MB
	3490E	N/A	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
	3590-1*	N/A	MEDIA3/4	(R/W)	128	10GB, 20GB
	3590-E** (3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10GB, 20GB 20GB, 40GB
STAND ALONE	3480	N/A	MEDIA1	(R/W)	18	200MB
	3490	N/A	MEDIA1	(R/W)	18	200MB
	3490E	N/A	MEDIA1 MEDIA1/2	(R) (R/W)	18 36	200MB 400MB, 800MB
	3590-1*	N/A	MEDIA3/4	(R/W)	128	10GB, 20GB
	3590-E** (3490E emulation)	N/A	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10GB, 20GB 20GB, 40GB
	3590-E** (3590-1 emulation)	N/A	MEDIA3/4 MEDIA3/4	(R) (R/W)	128 256	10GB, 20GB 20GB, 40GB

Table 4. Tape Storage Configurations (continued)

Library Model	Subsystem Device Type	Library Attachment	Media Supported	Recording Technology	Noncompacted Data Capacity
Note: MB = 1 048 576 bytes GB = 1 073 741 824 bytes (R) = Read only (R/W) = Read and write MEDIA1 = IBM Cartridge System Tape MEDIA2 = IBM Enhanced Capacity Cartridge System Tape MEDIA3 = IBM High Performance Cartridge Tape MEDIA4 = IBM Extended High Performance Cartridge Tape *3590-1 represents the 3590 Model B Tape Subsystem and is a system defined esoteric. **3590-E represents the 3590 Model E Tape Subsystem and is not a system defined esoteric. It is supported in a library as a 3590 Model E tape subsystem rather than what it is emulating. For information concerning cartridge storage feature options and cartridge capacities for these tape devices, refer to *TotalStorage Automated Tape Library (3494) Introduction and Planning Guide*, *3480 Magnetic Tape Planning and Migration Guide*, *3490 Models A01, A02, A10, A20, B02, B04, B20, and B40 Introduction*, *3490 Planning and Migration Guide*, and *3590 Introduction and Planning Guide*. The library models indicated can be configured with any combination of correlating tape subsystem devices. These configurations may vary in the number of drives, slots, and media type supported in the libraries. OAM is not aware of the type and number of channel attachments used to connect the supported Magnetic Tape Subsystems to the ESA/370 or ESA/390® processor. Any number and type (serial or parallel) of channel attachments supported by these subsystem configurations and the processor to which they are attached may be used.					

For planning purposes, use Table 4 on page 37 to determine the estimated capacity of an IBM cartridge system tape when OAM is storing either objects or the backup copies of objects on IBM cartridge system tape written on IBM tape subsystems.

The capacity of an IBM cartridge tape written by OAM and containing the primary or backup copies of OAM objects can be affected by a variety of factors, such as:
- The size of the object being stored.
- The use of compaction, and whether the data OAM is storing is already hardware or software compacted.
- Whether tape compaction is used or not.
- The tape volume percent-full specified for the Object or Object Backup storage group using the SETOAM statement in the CBROAM*xx* PARMLIB member.
- The tape full threshold specified for the Object or Object Backup storage group using the SETOAM statement in the CBROAM*xx* PARMLIB member.

If the object being stored is relatively small (16 kilobytes or less), then the capacity of the tape cartridge can be substantially reduced. Likewise, if the size of the object being stored on tape cartridges is large, the capacity of the tape cartridge can be increased and better utilized. This is due to the amount of buffer space needed between objects. The smaller the object size, the more buffer space is required to separate them.

Figure 11 on page 39 shows the approximate capacity of the following tape cartridges:
- IBM Cartridge System Tape and Enhanced Capacity Cartridge System Tape written in 18-track format (on IBM 3480 or 3490 base models)

- IBM Cartridge System Tape and Enhanced Capacity Cartridge System Tape written in 36-track format (on IBM 3490E enhanced capability models)

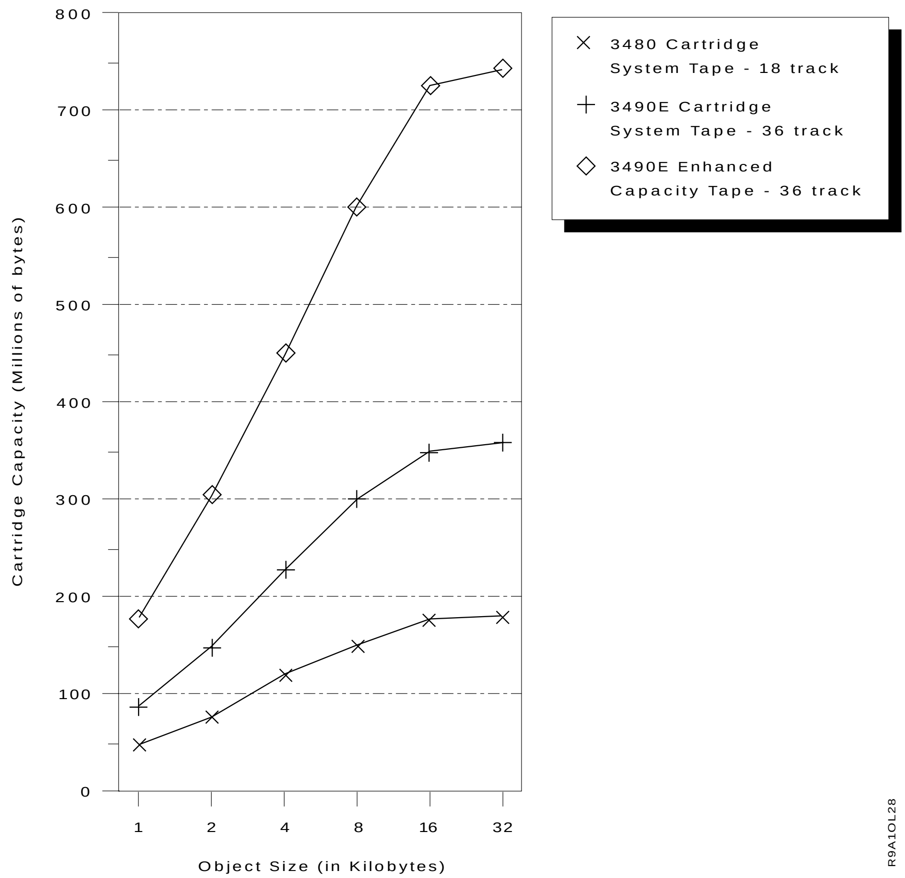

Figure 11. Tape Cartridge Capacity Versus Object Size

The tape compaction capability provides hardware compaction in the tape control unit and can increase the effective capacity of the tape media. It is recommended the compaction feature be enabled when OAM is writing primary copies or backup copies of objects to tape. To enable tape compaction, perform one of the following tasks as appropriate:

- Specify the TAPECOMPACTION keyword on the SETOAM statements in the CBROAM*xx* PARMLIB member.
- Omit the TAPECOMPACTION and the NOTAPECOMPACTION keywords on the SETOAM statements in the CBROAM*xx* PARMLIB member and specify a DATACLASS on the SETOAM statement. In the definition of the data class (specified with the DATACLASS keyword on the SETOAM statement), specify a COMPACTION option of "YES".
- Omit the TAPECOMPACTION and NOTAPECOMPACTION keywords on the SETOAM statements in the CBROAM*xx* PARMLIB member and do not specify a DATACLASS keyword on the SETOAM statements. Instead, specify the COMPACT=YES option in the DEVSUP*xx* PARMLIB member processed during MVS IPL.

If the data OAM is storing already is compacted, you should not expect any increase in the effective capacity of a tape cartridge due to the use of compaction. This is true in the case of image data (such as ImagePlus® algorithm suited for image data). In addition, if the application invoking OAM for storing data is compacting the data, such as with the Item Access Facility (IAF) program product, an increase in the effective capacity of a tape cartridge through the use of compaction should also not be expected.

OAM provides the capability for each Object or Object Backup storage group to specify what percent full OAM is to fill each tape cartridge belonging to the storage group. This option is specified with the TAPEPERCENTFULL keyword on the SETOAM statement in the CBROAM*xx* PARMLIB member.

If the installation chooses to have OAM fill the tape cartridges to a certain percentage of their estimated capacity, the approximate capacities listed in Table 4 on page 37 should be reduced. For example, if the installation specifies the tape volumes should be filled to 90% of the estimated capacity, the approximate capacities listed in the two prior tables should be reduced by 10%.

Tape Volumes

Each tape volume is identified by a unique volume serial number. The volume serial number must conform to MVS volume serial number naming conventions as documented in the *z/OS MVS JCL Reference*. The serial number on the tape volume cannot match the serial number assigned to any other SMS tape, DASD, or optical volume within the entire storage environment of the customer. The serial number must be unique within the installation.

Tape volumes can be used either in automated or manual tape libraries, or with stand-alone tape drives. For information concerning mounting, demounting, entering, and ejecting tape volumes into tape libraries or information regarding tape cartridges (considerations, capacities, and planning for their usage), refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Tape volumes reside in a protective housing known as a *tape cartridge*. The following types of tape media can be housed within a tape cartridge and used on IBM tape drives:
- IBM Cartridge System Tape (MEDIA1)
- IBM Enhanced Capacity Cartridge System Tape (MEDIA2)
- IBM High Performance Cartridge Tape (MEDIA3)
- IBM Extended High Performance Cartridge Tape (MEDIA4)

When objects are stored on tape volumes through an OSREQ STORE request, they are assigned to a specific Object storage group. OAM selects the appropriate tape cartridge type based on the DATACLASS parameter (if applicable), or the TAPEUNITNAME parameter that is specified for the storage group on the SETOAM statement. If the tape volume is allocated for a stand-alone tape drive, the TAPEUNITNAME determines the device type to be used, so the characteristics of the tape cartridge must be consistent with the capabilities of the tape drive. See Table 58 on page 470 for a diagram of the process of storing an object to tape.

Tape Volume Types

Three types of tape volumes are associated with object tape support:
- A group volume that is associated with an Object storage group
- A backup volume that is associated with the Object Backup storage group
- An MVS scratch volume that, when added dynamically, appears as belonging to an Object or Object Backup storage group when another volume is needed by OAM to satisfy a request

Group and Backup Volumes

Group and backup volumes are volumes that are already assigned to an Object or Object Backup storage group. If there is a request to write objects to a tape volume that is already assigned to an Object or Object Backup storage group, the tape volume that OAM selects must have sufficient space available to satisfy the request. If there are no tape volumes with sufficient space to satisfy the request, a scratch tape volume is allocated to handle the write request.

Scratch Volumes

When a scratch tape is added to an Object or Object Backup storage group, it is assigned the TAPEUNITNAME of the SETOAM that is statement currently associated with the storage group. Even though there is a tape unit name specified for the group, the ACS routines (for environment ALLOC) can override this tape unit name specification by assigning the allocation to a Tape storage group that directs the allocation into an ATLDS or a MTL. For example, the SETOAM statement can have a TAPEUNITNAME of 3480 associated with it. When the ACS routine runs for environment ALLOC, however, it overrides the information on the SETOAM statement and allocates the scratch tape to reside in an automated or manual tape library (neither can contain a 3480 device). In this case, the TAPEUNITNAME is automatically overwritten with the exact device type that was used to first mount the tape volume when it was first added to the storage group. This ensures that the volume is allocated to a compatible tape device after it is ejected from the tape library dataserver.

Format of the Object Data on the Tape Media

OAM records object data on tape volumes using the BSAM OPEN, WRITE, CHECK, NOTE, POINT, SYNCDEV, and CLOSE macros to process the data recorded.

If the tape volume is a primary volume, belonging to an Object storage group and containing the primary copy of the objects, the data set name of the physical sequential data set is **OAM.PRIMARY.DATA**. Since the same data set name is created on multiple OAM tape primary volumes, the data set is not cataloged.

If the tape volume is a backup volume belonging to an Object Backup storage group, and it contains the first backup copies of objects, the data set name of the physical sequential data set is **OAM.BACKUP.DATA**. If the tape volume is a backup volume belonging to an Object Backup storage group, and it contains the second backup copies of objects, the data set name of the physical sequential data set is **OAM.BACKUP2.DATA**. Because OAM creates the same data set names on multiple OAM tape backup volumes, it does not catalog the data sets.

Note: Allowing the ACS routines to assign or change the data class assignment of a tape volume is not recommended. The data class for tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information either at the storage group level or at the OAM global level and best suits the requirements for the tape volume being allocated. Allowing the ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). It is the installation’s responsibility to ensure that their ACS routines are written so that they do not alter the data class construct for OAM tape volumes.

NOT Programming Interface information

Each user object is recorded as one or more records within the data set. The maximum number of user object bytes within a single record is 32 628. No record contains data from more than one user object. Each record containing object data is self-describing and contains a 128-byte prefix. The 128-byte prefix contains the following information:
- Prefix identifier
- Prefix version number
- Prefix length
- Collection name
- Object name
- Offset of first byte of user data contained in this record
- Length field containing number of bytes of user object data in this record
- Reserved space

When an OSREQ STORE macro is issued in order to store an object and that object is to be written to tape, OAM causes the object data to be physically written to the tape media, prior to the OSREQ STORE macro returning control to the application program.

Note: A single object never spans tape volumes.

End of NOT Programming Interface information

Tape Drives

In addition to optical disk drives, OAM also provides the ability to store the primary copy, or the backup copy, or both, of objects on tape volumes that can be mounted on these tape drives. OAM provides support for various IBM tape subsystems (stand-alone tape drives), the automated tape library dataserver (ATLDS), and the manual tape library (MTL).

Unlike optical drives, tape drives are not defined to the system through ISMF. The system allocates the tape drives to use to satisfy read and write requests of objects. The system relies on information from the ACS routines, and the location of the volume to be mounted to determine what device should be allocated to handle the request. If the volume is a library-resident volume (residing in an ATLDS or MTL), the system chooses a device to satisfy the request. If the volume resides outside of an ATLDS or MTL, the system allocates a stand-alone drive. The drive selected for use with a stand-alone tape depends on the TAPEUNITNAME associated with that tape in the TAPEVOL table row. If this is an MVS scratch tape (which has no TAPEVOL table row yet), the TAPEUNITNAME associated with the storage group to which the tape is assigned determines the type of stand-alone device which is allocated.

OAM supports automated and manual tape libraries and various models of IBM tape subsystems that attach to an IBM ESA/370 or ESA/390 capable processor and are supported by any OS/390 or z/OS release level operating system. See Table 4 on page 37 for detailed information on all supported models. For more information concerning tape hardware configurations, and OAM's role with the tape library dataservers and stand-alone tape drives, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Dynamic Allocation of Tape Drives

OAM uses the MVS dynamic allocation macro (DYNALLOC) for all tape drive allocations. Tape drives remain allocated while OAM is using them, but are then dynamically deallocated when OAM no longer needs them. Thus, the tape drives are obtained and released as OAM needs them; the tape drives are not dedicated to OAM. The tape drives are allocated to the OAM address space and not the application address space (for example, CICS, IMS MPP, or IMS BMP) invoking the OAM application programming interface (the OSREQ macro).

The allocated device could be a stand-alone tape drive or a drive inside of an automated or manual tape library. The device allocation depends on:
- The contents of the ACS routines
- Whether this is a mount for an existing OAM tape containing objects or a mount for an MVS scratch tape
- The location of the volume to be mounted

Although tape drives are not permanently allocated to the OAM address space, tape drives must be available to the OAM address space when they are needed to handle a request to store or retrieve an object on a tape volume.

The maximum number of tape drives capable of being concurrently allocated to the OAM address space is controlled by the sum of the MAXTAPESTORETASKS and the MAXTAPERETRIEVETASKS keywords at the OAM global level on the SETOAM statement.

Failing Dynamic Allocation and MVS Allocation Recovery

If the initial dynamic allocation of a tape drive fails, then OAM retries the dynamic allocation every 10 seconds for a full minute. If after one minute OAM does not successfully allocate the required device, OAM issues message CBR6425I indicating to the operator that OAM has not allocated a tape drive. The CBR6425I message lists the object name, collection name, storage group name, and tape volume name (SCRTCH if for an MVS scratch allocation) for this tape allocation request.

OAM continues to retry dynamic allocation every 10 seconds, for another four minutes or until a suitable tape drive is allocated, whichever comes first. During this period of time (up to five minutes) that OAM is trying to allocate a tape drive, MVS Allocation Recovery processing is disabled and OAM is retrying the dynamic allocation by reissuing the dynamic allocation macro.

If OAM does not successfully allocate a suitable tape drive at the end of five minutes, then OAM reissues message CBR6425I along with message CBR6400D. The CBR6400D message lists the storage group name and tape volume name for this tape drive allocation request and asks the operator if OAM should continue to retry or cancel the request to dynamically allocate a tape drive. If the operator replies “R” (meaning retry) to the CBR6400D message, OAM again issues the dynamic allocation macro, but with MVS Allocation Recovery processing enabled. If the allocation request cannot be satisfied immediately, MVS Allocation Recovery issues message IEF238D and no other dynamic allocations, dynamic deallocations, OPENs, or CLOSEs can occur in the OAM address space until this allocation completes or is canceled. If the operator replies **C** (meaning cancel) to the CBR6400D message, OAM fails the tape drive allocation and its associated OAM request. Any other reply to the CBR6400D message causes OAM to reissue the CBR6425I and CBR6400D messages.

The actions performed during MVS Allocation Recovery processing are affected by the options specified in the ALLOC*xx* member in PARMLIB. If an eligible device is not made available to OAM, the dynamic allocation request fails and the associated store or retrieve request for the object also fails. For more information concerning the ALLOC*xx* member of PARMLIB and the installation defaults for handling allocation requests, refer to *z/OS MVS Initialization and Tuning Reference*.

Note: This processing applies to stand-alone devices as well as devices inside of automated or manual tape libraries. Issuance of the message and the usage of the timer logic are not restricted to nonautomated environments.

Retrying or Canceling a Volume Mount

If OAM is waiting for the mount of a volume after the appropriate device has been allocated and the five minute default or the time specified on the MOUNTWAITTIME parameter of the SETOAM statement has elapsed with no mount occurring, message CBR6405D is issued to the operator to ask if the mount should be retried or canceled. If the operator replies "R", then the mount message is left on the console until the installation-specified amount of time (MOUNTWAITTIME) has again elapsed or the mount has been completed. This process continues until the requested volume is mounted or until the operator replies **C**. Should the operator cancel the request, one or more of the following actions will occur:

- Message CBR2003I, stating that the tape volume that was requested to be mounted was not found, and is marked lost, is issued.

  Note: This message is not issued for an MVS scratch tape mount that is used to satisfy the request. This message is only issued for tape volumes that have rows in the TAPEVOL table.
- The volume is marked lost so that future specific requests for this volume immediately fail with a nonzero return and reason code. However, the volume is only marked lost if it is a volume with a row in the TAPEVOL table. Although future specific requests for this volume fail, the nonspecific requests are attempted using a different volume (as long as this was not a mount scratch that is canceled).

  Note: To clear the status of a lost volume, use either of the following commands:

  ```
  F OAM,UPDATE,VOLUME,VOLSER,LOSTFLAG,OFF
  ```

  or the

  ```
  F OAM,RESTART
  ```
- Specific requests that have been submitted to OAM for the lost volume fail with a nonzero return and reason code.
- If the request that required the mount was a specific request, it fails with a nonzero return and reason code.
- If the request that required the mount was a nonspecific request with a corresponding row in the TAPEVOL table, OAM tape volume selection attempts to find another volume for the request. If the cancellation was for an MVS scratch tape, OAM fails the request.

Compatibility of Devices and Data Format for Object Retrievals

Whenever an object exists on tape, OAM can retrieve the object only when a device compatible with the format of the data written on the tape volume is available at the time of the retrieve request. Also, in order to retrieve any objects from tape, it is necessary that OAM is initiated with a valid CBROAM*xx* specification in order for object tape support to be in effect.

If an Object or Object Backup storage group that used to have its objects written on tape is now having objects written to optical media, those objects can also be read back using a tape device compatible with the format of the data written on the tape volume available at the time of the retrieval request. In order to read data back from tapes previously written in a group that is no longer writing data to tape, there *does not* have to be a SETOAM statement for that group in the CBROAM*xx* PARMLIB member processed.

OAM has a default of one system read and one system write task; each group has a default of one read task and one write task. The installation should ensure that there is a compatible tape device available for allocation at the time the retrieval request is received. If there is no device available for allocation, MVS Allocation Recovery issues allocation recovery messages, requesting that an offline or inaccessible device be made available (see “Dynamic Allocation of Tape Drives” on page 42 for more information). If this is not possible, the retrieval request for the pending mount fails.

Tape Libraries

Tape libraries consist of a set of tape volumes and the set of tape drives on which those tape volumes can be mounted. A tape library can consist of one or more IBM 3490, 3490E, and 3590 subsystems. These drives are configured into automated or manual (3590 drives are not supported in MTLs) tape library dataservers that contain library-resident tape volumes. The storage administrator defines tape libraries to SMS using ISMF library management definition panels (refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for more information on how ISMF is used in tape library management). A tape library can contain tapes from multiple storage groups and a storage group can span up to eight libraries (ATLDSs, MTLs, or a combination of these).

Chapter 2. Planning for OAM

In many ways, planning is the most important phase of the OAM product's implementation and administration cycle. Time spent in planning is fully repaid in time, effort, and money saved by a well-implemented installation and a smooth transition to full system integration. This chapter identifies key areas that must be addressed during planning. Rather than repeat large amounts of information available elsewhere in the DFSMS and storage management libraries, this chapter focuses specifically on object-related issues and provides references to other resources. A case study, included at the end of this chapter, illustrates how planning concepts can be applied in a typical situation.

Setting Up the Planning Team

To most effectively use the information in this chapter, your planning team should include individuals with significant technical expertise in the following areas:

- **Data Facility Storage Management Subsystem-Related Products.** OAM is a component of DFSMS and as such interacts closely with other members of the DFSMSdfp family of products. Knowledge of system-managed storage (SMS) and the interactive storage management facility (ISMF) are essential for successful planning and implementation. Additionally, familiarity with DFSMShsm and the other DFSMS components provides a meaningful context for understanding OAM.
- **DB2.** OAM object support uses DB2 databases to store internal information (such as object indexes) and objects. Implementing OAM object support is likely to have a significant impact on your installation's DB2 space requirements. If OAM is to be set up in a Parallel Sysplex environment, DB2 data sharing installation and knowledge is also required.
- **Integrated Catalog Facility (ICF).** OAM uses collections to subdivide object data within Object storage groups. Collections must be cataloged in integrated catalog facility catalogs.
- **Customer Information Control System (CICS).** If OAM is invoked by CICS transactions, the planning team will need to evaluate the effect of their interaction.
- **Hardware Configuration Definition (HCD).** HCD is used to define devices to the hardware configuration.
- **Information Management System (IMS).** If OAM is invoked by IMS transactions, the planning team will need to evaluate the effect of their interaction.
- **Time Sharing Option (TSO).** If OAM is invoked by TSO transactions, the planning team will need to evaluate the effect of their interaction.
- **Cross-system Coupling Facility (XCF).** If OAM is to be established in the Parallel Sysplex environment, the planning team will need to evaluate the impact to the coupling facility requirements and resources.

As with any major installation, the OAM planning effort should also involve people with project management experience and representatives of the end-user areas that will be affected by the implementation.

Undergoing the Planning Process

This chapter is organized into various sections, one for each phase of the planning process, and one for the case study. The following subjects are covered:

Analyzing Your Business Environment

Your processing environment reflects the unique goals, procedures, and structure of your business; therefore, successful OAM object support implementation must be grounded in a thorough analysis of that business environment. The technical installation can then evolve logically from the functions and requirements you have defined.

The first task in the analysis process is to characterize the objects that will be processed. Among the most useful classification categories are:

- **Size.** Are objects small, medium, or large? What are the criteria for these categories in your installation?
- **Activity.** How often are objects retrieved? How often are new objects stored? Is one object accessed many times or are many objects accessed one time each? What is the required response time for accessing an object?
- **Volume.** How many objects of each size will be created? How many objects must be processed every hour or every day?
- **Life cycle.** Is the activity level stable or does it change in response to a business cycle (such as monthly billing)? Are such changes random or periodic? How frequently do these changes occur? Are objects backed up? How long must objects be retained? Do you plan to delete objects automatically?
- **OAMplex.** If you are planning to run an OAMplex, how many systems (OAMs) will be in your OAMplex? Which systems will have the hardware physically attached to assist in determining where OSMC should be run? How will the storage group and library disbursement be handled between systems?

As a result of this analysis, you will be able to:

- Determine criteria for grouping objects
- Establish performance objectives
  - Determine the best system for OSMC processing per storage group
  - Determine hardware distribution based on demand and location
- Identify storage management cycles

This analysis, in turn, leads you to create storage groups, collections, storage classes, data classes, and management classes through which OAM and SMS can implement your storage management policy. The ultimate goal is to develop a set of Storage Management Subsystem (SMS) constructs that you can use to accurately describe and respond to the complex reality of your business environment.

Grouping Objects

During the process of characterizing your installation’s objects, you probably discovered that the objects can be grouped in various ways. OAM uses the following techniques to group objects physically and logically:

- Object storage group assignment represents the physical storage, managed by OSMC according to your storage management policy.
- Collection assignment represents a logical relationship between objects.

Every object belongs to a collection; every collection belongs to an Object storage group. Each Object storage group can contain one or more collections; however, a collection can never span multiple Object storage groups.

Storage Groups

The Object storage group construct makes it possible to manage a set of storage devices as a single object storage area. Each Object storage group encompasses several types of storage devices to support an object storage hierarchy (as defined on page 11).

You can need to organize storage into physically separate groups, such as:

- Business needs
- Accountability
- Security
- Application isolation
- Device characteristics
- Connectivity

Collections

A collection typically contains objects that are used by the same application or are of a similar type. Collections are useful for dealing with sets of objects that are too large to be handled as a single object, but too small to warrant a separate Object storage group. For example, all objects in a collection can have the same default initial storage class and management class attributes.

You can organize objects into collections for a variety of reasons. For example, if objects related to a corporate division are kept in one Object storage group, it may be desirable to subdivide that Object storage group into collections of departmental data.

Establishing Performance Objectives

The object characterization process generally reveals that different response times are required for different sets of objects and that some objects are accessed more frequently than others. OAM uses the storage class construct to specify object performance objectives and availability requirements to SMS. Every object in the object storage hierarchy must have an associated storage class. The fact that every object has an associated storage class makes every object, by definition, SMS-managed.

Your business needs provide the service-level criteria on which storage classes are built. Table 5 on page 50 shows how you can specify performance objectives for different storage classes depending on the service levels required:

Table 5. Storage Class Service Levels for a Variety of Business Needs

Business Need	Service Level
Daily operation	Fast response; frequent access
Online customer inquiries	Fast response; occasional access
Quarterly batch processing	Medium response; periodic access
Legal retention requirements	Slow response; very infrequent access

A storage class does not represent any physical storage. OAM analyzes the storage class parameters and tries to meet the performance objectives by placing the object on a device that best meets those objectives. Using storage classes to force use of a specific device type can defeat the purpose of system-managed storage and cause serious inefficiencies. For example, using a storage class that causes objects to be written directly to optical media without being staged through DASD can degrade system performance and significantly increase the number of optical disks needed per day, because of the inefficient storing of optical volume table of contents (VTOC) information.

Consider separating the storage classes that are used to control objects for one application from the storage classes that are used for other applications. If it becomes necessary to change the performance objectives for objects used by an application, its associated storage classes can be changed without affecting the other applications.

For a detailed discussion of storage classes and how to plan for them, refer to *z/OS DFSMSdfp Storage Administration Reference*.

Identifying Management Cycles

Every business is subject to operational cycles that influence work flow. These cycles often have a direct effect on performance and availability requirements. The management class construct, in conjunction with storage classes and ACS routines, makes it possible for SMS to respond to these cycles as it manages object storage (see Figure 12 on page 51 for a representation of this process). Every object in the object storage hierarchy must have an associated management class.

As you analyze your business environment, consider the potential effects of these cycles on your work load and, therefore, your object access requirements:
- Accounting
- Reporting
- Manufacturing
- Marketing
- Backup
- Retention
- Physical location

Remember to factor into your analysis the frequency of each cycle (such as daily, weekly, monthly, quarterly, or annually).

To fully exploit the management class construct, it is necessary to understand class transitions and storage management cycles. A *class transition* is a change in an object's management class or storage class when an event occurs that brings about a change in an object's service level or management criteria. Class transition criteria are specified in management class definitions. For example, a management class might specify that 180 days from an object's creation date, the ACS routines should be invoked to determine if a class transition is needed.

Understanding Storage Management Cycles

A class transition occurs during a storage management cycle. A *storage management cycle* is an invocation of the OAM Storage Management Component (OSMC) for an Object storage group. The storage management cycle ensures that every object that is scheduled for processing is placed in the correct level of the object storage hierarchy (as specified by its storage class), is deleted or backed up (as specified by its management class), and, if necessary, is flagged for action during a subsequent storage management cycle.

Note: OSMC does not perform any storage management processing against the Object Backup storage groups.

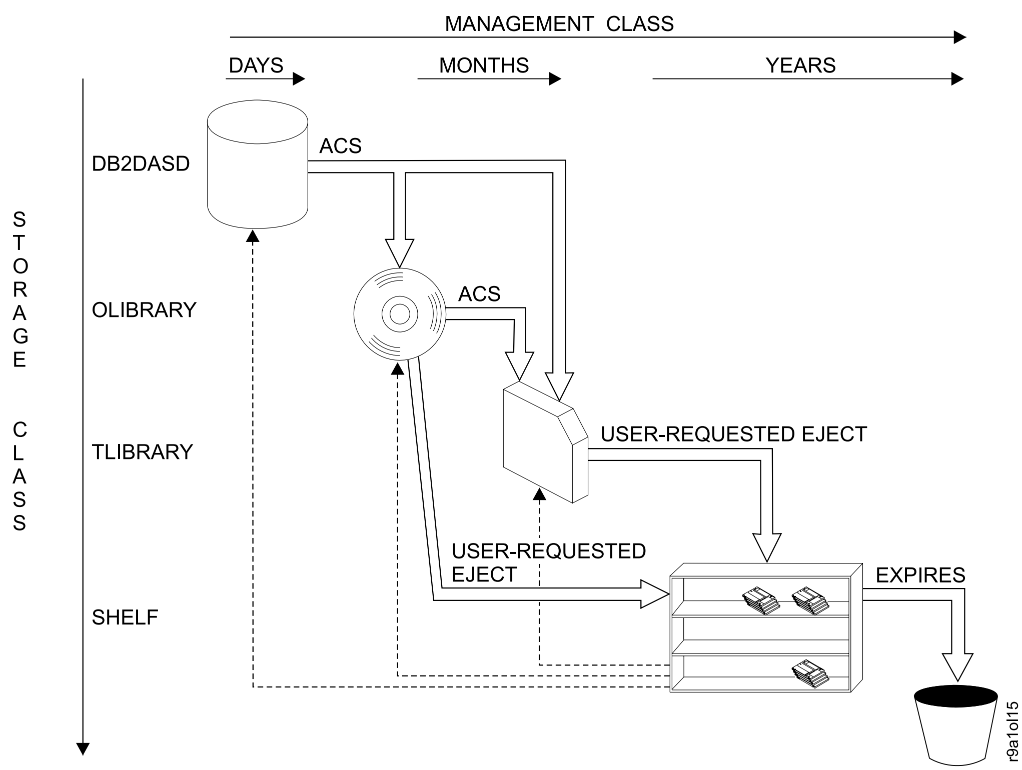

Figure 12. Example of the Concept of Using Class Transitions and ACS Routines to Change Management Classes

Note: OAM does not support the expiration of tape volumes, but it does support the expiration of objects residing on the tape volumes. For more information concerning tape volume expiration dates and how to avoid expiring tape volumes, see the discussion on page 52.

Storage management cycles can run automatically once each day, starting at a period of time that you specify (for example, during third shift). If you are in an OAMplex and using automatic startup, you should also specify a system name; otherwise, multiple systems will try to process the same storage group at the same time. As it executes, OSMC checks to see if an object is scheduled for processing. An object is scheduled for processing for any one of the following reasons:

- It was stored since the last storage management cycle.

- It was retrieved and the UPD=N parameter is not specified on the OAM1 statement in the IEFSSN*xx* member of PARMLIB.
- Its storage class, management class, or expiration date has been changed since the last storage management cycle.
- It was marked for a class transition by a previous storage management cycle.
- It was marked for expiration by a previous storage management cycle.

Checking for Object Deletion

When OSMC encounters an object in the Object storage group that is scheduled for processing, it first checks to see if the object has been marked for expiration. If it has, OSMC requests the Library Control System to delete it, with the approval of the auto-delete installation exit (see "Auto-Delete Installation Exit (CBRHADUX)" on page 509).

See "Objects Not Selected for Expiration Processing by OSMC" on page 205 for information on handling expired objects on volumes in the Object Backup storage group.

If the object is being deleted from an optical rewritable volume, the deleted space and deleted counts are updated, and the object name, collection name, volume serial number, and sector location are added to the deleted objects table for later physical deletion. If the object is deleted from a tape volume, the tape volume record is updated with the deleted kilobytes.

Each time OAM receives a request to delete an object from a tape volume, OAM updates the number of logical kilobytes deleted from that tape volume by adding the size of the object which was just deleted to the existing logical kilobytes deleted value for that tape.

Expiration of OAM tape volumes is not supported. A volume expiration date for tape volumes is calculated using the same process for calculating the expiration date for optical volumes. This expiration date is stored for each tape volume in the TAPEVOL table. It is the user's responsibility to ensure that the tape management system used *does not* expire OAM tape volumes. Tape volumes are expected to continue to be available for use by OAM. One way to guarantee this expectation is to use the expiration date or retention period option on the DATACLASS parameter, and then specify DATACLASS on the SETOAM statement for the Object or Object Backup storage group to which the tape volume belongs. This expiration date is written in the header label on each tape added to the Object or Object Backup storage group whose DATACLASS specified the expiration date or retention period option.

Note: If DFSMSrmm is being used, it is suggested that the following vital record specifications, shown in TSO format, be used for OAM-owned tapes:

```
RMM ADDVRS DSNAME('OAM.PRIMARY.DATA') COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('OAM.BACKUP.DATA') COUNT(99999) LOCATION(HOME)

RMM ADDVRS DSNAME('OAM.BACKUP2.DATA') COUNT(99999) LOCATION(HOME)
```

Determining Class Transition and Backup Requirements

If the object has not been marked for expiration, OSMC determines if a class transition is needed. When a class transition is indicated, OSMC invokes the ACS routines; these routines evaluate the object's class assignments and change them if necessary. Next, OSMC performs any actions that are indicated by an object's class assignments:

- OSMC interrogates the storage class to determine if the object should be placed at a different level of the object storage hierarchy.
- OSMC checks the management class to set the *next* date on which management action (that is, expiration or class transition) will be needed.
- OSMC makes up to two backup copies of the object if the management class indicates that one or two backup copies are required, and the requested number of backup copies does not currently exist.

This situation can result from any of the following scenarios:

- This is the first storage management cycle for a new object.
- An application change request has changed an object's management class to one that requires backups.
- A class transition has changed an object's management class to one that requires backups.
- An application change occurs so that the object's management class requires two backup copies where currently only one backup copy exists.
- A class transition occurs so that the object's management class requires two backup copies where currently only one backup copy exists.

For objects with a primary copy on DASD, tape, or optical, OSMC performs the following actions:

- OSMC makes backup copies in accordance with the management class that is assigned to the object. Backup copies can be directed to an optical disk, tape volume, or both, depending on the definitions that are associated with the Object storage group on which the primary copy of the object resides.

  If the management class requires a single backup copy, OSMC directs the backup copy to the associated Object Backup storage group. This storage group can be located on either an optical disk or a tape volume. If the management class indicates that two backup copies of the object are required and the SETOSMC statement associates two Object Backup storage groups with the Object storage group where the primary object copy resides, the first backup copy is written to one Object Backup storage group and the second backup copy is written to the other Object Backup storage group. These Object Backup storage groups can reside on the same or different media types (optical or tape).

  For backup copies to be made to tape volumes, there must be SETOAM statements with properly specified STORAGEGROUP and TAPEUNITNAME parameters for the Object Backup storage groups.
- The object is presented to the ACS routines in the CTRANS environment to allow for class transition.
- The next scheduled processing for the object is determined. If prior to the expiration for the object, OSMC processing for the object is next scheduled for when the object is to expire (based on the OSREQ STORE/CHANGE specifications for the RETPD and the management class assigned to the object).
- On expiration of the object, OSMC deletes all copies of the object (primary and all the backups). Deletion removes information from the object directory but may or may not result in physical deletion, depending on the type of media. OSMC does not physically delete object copies residing on tape and WORM media. For objects that reside on tape volumes, the number of logical kilobytes that are deleted from the volumes is incremented for each object deleted.

Developing Appropriate Management Classes

Like storage classes, management classes can be developed to meet a variety of needs. Develop as many management classes as necessary to support the class transitions and storage management cycles that are required by your business. As you attempt to develop management classes, work through the following steps:

1. Analyze your applications to plan for large groups of objects that have the same management requirements. You can assign these objects to one management class and store them together in one collection.
2. Identify objects that are good candidates for early deletion or class transition.
3. Separate objects that do not need to be backed up from those that do.
4. Identify objects that require a delayed class transition.
5. Identify objects with medium to low response-time requirements, so that they can be moved to tape or optical storage as soon as possible.

Note: To avoid processing inefficiencies or unexpected results, or both, careful analysis of the end results of class transition is critical for a successful implementation.

For a detailed discussion of management classes and how to plan for them, refer to *z/OS DFSMSdfp Storage Administration Reference*.

OSMC Processing Management in an OAMplex

If an Object storage group has an OSMC processing system name, and the timer in the OSMC cycle start/stop time window is triggered, the storage group processing is initiated from the specified system. If the timer is triggered and the Object storage group does not have an OSMC processing system name specified, storage group processing is initiated on all OAMs within the OAMplex. **This causes DB2 and OSMC errors, so specifying the system name is important in this environment.**

If using the F OAM,START,OSMC command to start a full OSMC cycle, the storage groups with an OSMC processing system name that matched the system where the command was entered will be started. Also, storage groups with no specified OSMC processing system name will be started.

A specific F OAM,START,STORGRP,*storage_group* command is always honored on the system where the command is issued, even if another system name is specified in the processing system name. It is recommended that you try to localize the OSMC processing to the system where the hardware is physically online for that storage group. This reduces the amount of cross-system processing that is required. Also, if multiple systems are running different storage groups at a time, the impact to DB2 should be analyzed (especially if normal activity to the storage group is occurring at the same time).

Concluding the Business Analysis Phase

As you have seen, OAM and SMS support a variety of conceptual structures through which you can describe your business environment and specify a storage management policy. Furthermore, each of these structures offers a significant amount of flexibility.

Unless your business environment is an unusually simple one, you should expect the analysis process to require several iterations. There are likely to be several equally viable ways to define your Object storage groups and classes. Unfortunately, there are no magic algorithms for choosing which approach to implement. That decision can be made only by one who knows the most about your business: *you*.

At this point in the planning process, you should have a rough idea of how your objects will be organized into Object storage groups, collections, storage classes, and management classes. The next step is to analyze your processing environment.

Analyzing Your Processing Environment

Installing a new product is rarely an isolated event. Planners must evaluate how the existing environment will be affected by the new product, as well as how the new product must be customized to integrate with the existing components. This section presents guidelines for analyzing the hardware and software that make up your processing environment.

Hardware

The use of OAM presupposes a considerable mainframe environment. To take advantage of the full range of OAM capabilities, the environment should provide substantial amounts of internal and external storage. In addition to standard direct access storage device (DASD) devices, OAM also supports the use of optical disk drives and tape devices inside of and outside of ATLDSs and MTLs within an object storage environment. OAM does not have any hardware prerequisites; however, you can augment internal and external storage capacities to accommodate an increased work load.

Grouping Tape Devices

Tape devices can be grouped together and be defined as one group to the system. For instance, a group of 3490 tape drives in the same room can be grouped together and defined as 3490GRP. These tape device groups are known as *esoterics*. Once an esoteric is specified on the SETOAM statement for a group, it is necessary to ensure the existence of that esoteric as long as there is an OAM tape which specifies that esoteric in the TAPEUNITNAME field of the TAPEVOL table. It is also imperative that the contents of that esoteric not be changed to introduce incompatible tape device types. Should the esoteric name be deleted or changed, the volumes associated with the esoteric name cannot be allocated. Because the TAPEUNITNAME cannot be resolved, the tape required for the request is not mounted and the allocation request fails.

Note: In the case of scratch allocations for an ATLDS or MTL, the esoteric TAPEUNITNAME associated with a storage group is overridden with the exact device type for the device that is allocated for the MVS scratch tape mount.

Considering Storage Configurations

Use of optical or tape storage is *not* required for OAM. In fact, as you first begin to work with objects, it can be desirable to implement a pilot application that uses only DASD. If and when optical or tape storage is included in your storage management scheme, one or a combination of the hardware configurations described in Table 3 on page 25 and Table 4 on page 37 can be used as a standard configuration. For more information and examples concerning standard tape library hardware configurations, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Deferring Hardware Ordering Until After Work Load Analysis

The task of ordering hardware should be deferred until after the OAM work load is analyzed and resource requirements are estimated (see "Estimating Resource Requirements" on page 57). Depending on the volume of objects to be processed and stored, you can order additional DASD, optical storage devices, tape devices, or all three. Remember that an Object or Object Backup storage group can specify no more than eight optical libraries. Be sure to request appropriate documentation when you place the hardware order (see Table 1 on page xiii).

For more information about OAM hardware-related issues, see "Appendix A. Sample Optical Hardware Configurations" on page 323, and refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Software

Software analysis must address two distinct types of programs:

- System software—system control programs, transaction control programs, security packages, communication programs, database management systems, storage management systems, and similar global system products
- Custom applications—locally-written application programs, customized exits, site-specific macros, and other software unique to your installation

Additionally, OAM has software prerequisites that you must install prior to implementing OAM.

System Software

OAM object support is designed for use in a z/OS environment. It is a component of DFSMS (the base for SMS) and uses DB2 extensively. The OSREQ application programmer's interface (API) to OAM can be invoked by CICS and IMS transactions, TSO programs, and MVS batch jobs. See *z/OS DFSMS OAM Application Programmer's Reference* for additional information on the OSREQ API.

Note: The following restrictions should also be considered when planning for the installation of OAM:
- Only one OAM address space can be active per MVS system.
- Only one DB2 subsystem application can be associated per OAM address space.
- Optical devices can only be directly accessed by the OAM address space where they are physically online; however, they can be indirectly accessed by other OAMs within an OAMplex.
- If in an OAMplex, the DB2 subsystems connected to the OAMs in the OAMplex must belong to the same DB2 data sharing group.

Custom Applications

By definition, custom applications are unique to your installation. ***The planning team is responsible for thoroughly investigating the installed software base to determine what, if any, custom application constraints apply for OAM implementation.***

Software Prerequisites

The following software is the minimum release level that must be installed to enable this release of OAM:
- z/OS Version 1 Release 3
- IBM DATABASE 2 Version 4 (with DB2 data sharing if installing an OAMplex)
- If the OSREQ macro is invoked from a CICS transaction, then IBM CICS/MVS® Version 2, Release 1.1 is required:
- If the OSREQ macro is invoked from an IMS application, then one of the following is required:
  – IBM IMS/VS Version 2
  – IBM IMS/ESA® Version 4 Database Manager

For more information about the software requirements for DFSMS, refer to *z/OS DFSMS Migration*.

Software can be ordered as soon as you have determined which, if any, of the required components must be added or upgraded. Remember to request supporting documentation when you place the software order (see Table 1 on page xiii).

If you are adding significant new system software along with OAM, consider phasing the installations. Install and test one product at a time; then, when the system is stabilized, add another product. This approach simplifies error diagnosis by limiting the number of potential problem areas.

Estimating Resource Requirements

To fully assess the potential impact of implementing OAM, you must translate the work load estimates from the business analysis phase into resource requirements. This section provides some general considerations for evaluating resource needs and formulas for estimating space requirements for DASD, tape, and optical storage.

After the estimating process is complete, you will be able to evaluate the capacity of the current processing environment and prepare to increase it, if necessary.

| **Attention:** The formulas, constants, and performance rates used in this section are only for estimating purposes. They do not constitute benchmarks or guarantees and are provided solely as guidelines to assist you in your planning effort. They should not be interpreted as estimates for operation under normal work loads.

General Considerations

Estimating resource requirements is more of an art than a science. Although rules of thumb and generalized algorithms are useful, each installation's needs and environment are unique. To optimize the estimating effort, each planning team should allow a reasonable margin of error and be alert to the subtle interactions that can affect performance, throughput, and, therefore, actual resource requirements. The following comments are included to stimulate your analysis during the estimation process:

- Object size and activity level influences virtually all aspects of resource utilization. The validity of the remaining resource estimates depends on the accuracy of these fundamental assumptions.
- When estimating object transfer rate and other throughput issues, remember that OAM is part of a complex system, influenced by many factors.
- Application design can have a significant effect on OAM efficiency. For example, careful use of buffering can reduce virtual storage requirements.
- When estimating the amount of time needed to complete a storage management cycle, consider the following:
  – All data movement resulting from storage class changes, whether caused by class transition or application request, is handled during the storage management cycle.
  – Movement from DASD to optical or tape is typically faster than the reverse.
  | – The frequency and volume of backup copies of objects that are made during the cycle can increase processing time.
  – The effective drive transfer rate is faster for a single large object than for multiple small objects.

- In addition to estimating obvious needs, such as DASD, tape, and optical storage space, consider possible requirements for resources that are less directly related to the OAM installation. For example, do you have enough of the following:
  – Trained support personnel, application developers, and end users?
  – User IDs with the correct access authority?
  – Tapes to process backups for the DB2 tables?
  – Shelf space for shelf-resident optical disks and library-resident tape volumes associated with manual tape libraries?
  – Shelf space for shelf-resident tape volumes used with stand-alone tape devices?
  – Customized transport classes (CTCs) for OAM XCF processing to decrease any possible impact on cross-system processing based on locality of hardware versus the system originating the request.
  – DB2 threads and locks to process concurrent access to tables during an OSMC cycle?

DASD Storage

Implementing OAM requires a significant amount of direct-access storage. DASD space is required for the object databases and for the OAM administration database. This section provides techniques for calculating DASD space requirements for these databases. See “DASD Resources” on page 71 for an example of how these computations are applied.

In addition to the space requirements detailed here, consider DASD space needed for the optical configuration database, OAM-related programs (system and application), and the catalog that contains collection name entries.

System Paging

The various subcomponents of OAM make significant use of virtual storage resulting in a high auxiliary storage requirement. In most instances, an installation uses one or more dedicated DASD volumes for paging so that there is sufficient auxiliary storage. If after system tuning there is low paging activity, you can allocate other low-activity data sets to a paging volume.

Attention: A minimum of 210 megabytes (290 cylinders on a 3390 DASD) of auxiliary storage is required for OAM local paging. Other applications that use OAM may require additional auxiliary storage.

Object Databases

Each Object storage group has one object database. Each object database requires eight separate VSAM linear data sets. These data sets are for the object directory and its three indexes, the small object table and its index, and the large object table and its index. (See Figure 101 on page 454 for a diagram of an object database.) These data sets are explicitly allocated to give you control over data set placement and size. This control allows you to take full advantage of your configuration to optimize system and DASD I/O performance.

Before attempting to estimate your DASD requirements, you must have completed the business analysis process by establishing the following:
- Object grouping for all objects to be handled by OAM
- Object sizes within each Object storage group
- Number of objects to be stored in DB2 databases for each Object storage group

Data Set Name Qualifiers: Table 6 on page 59 provides descriptions and data set name qualifiers for each of the required data sets in the database.

Table 6. Data Set Name Qualifiers and Descriptions for Each

Qualifier	Description
OSMDTS	Object Directory Table; information about the object
OBJDIRX1	Object Directory Index 1; cluster index (odcreats) into the object directory table
OBJDIRX2	Object Directory Index 2; index (pending action date) into the object directory table
OBJDIRX3	Object Directory Index 3; index (collection name/object name) into the object directory table
OSMOTS04	4 kilobytes Object Table; storage area for objects with a length less than or equal to 3980 bytes (small objects)
OBJT04X1	4 kilobytes Object Table Index 1; index (collection ID/object name) into the 4 kilobytes object table
OSMOTS32	32 kilobytes Object Table; storage area for objects with a length greater than 3980 bytes (large objects)
OBJT32X1	32 kilobytes Object Table Index 1; index (collection ID/object name/segment name) into the 32 kilobytes object table
Note: kilobytes = 1 024 bytes.	

Variables: Table 7 describes the variables used in the DASD formulas:

Table 7. Variables Used in DASD Formulas

Variable	Description
nt	Total number of objects stored within an Object storage group, which includes objects stored in the small object table, objects stored in the large object table, and objects stored directly on optical or tape storage
ns	Maximum number of objects stored in the small object table
nl	Maximum number of objects stored in the large object table
aos	Average object size
b	Number of 4-kilobyte pages per track on the device (see Table 8)
cb	Number of 32-kilobyte blocks per cylinder on the device (see Table 8)

Constants: Table 8 summarizes the constants that are related to device type.

Table 8. DASD Device Characteristics

Device Type	3390	3380
Pages per track (b)	12	10
Blocks per cylinder (cb)	22	19
Tracks per cylinder	15	15

Formulas: Table 9 on page 60 provides calculations for estimating DASD requirements for the eight object data sets.

These formulas do not include any significant free space. The formulas should be calculated using the absolute maximum number of objects anticipated, plus whatever additional free space you require.

The results of the formulas are expressed in tracks on direct-access storage, which can be converted to cylinders using the information in Table 8 on page 59. The exception is the large object table, where the result of the calculation is expressed in cylinders.

The results of all these calculations must be rounded up to the next higher integer.

Table 9. Example of Calculations For Determining DASD Requirements

OBJECT DIRECTORY TABLE or INDEX	CALCULATION
Directory table (OSMDTS)	*nt* ÷ (35 x *b*)
Directory index 1 (OBJDIRX1)	*nt* ÷ (260 x *b*)
Directory index 2 (OBJDIRX2)	*nt* ÷ (166 x *b*)
Directory index 3 (OBJDIRX3)	*nt* ÷ (68 x *b*)
4K object table (OSMOTS04)	If *aos* > 1900 bytes: (*ns* ÷ *b*) x 1.1 *If *aos* < 1900 bytes: (*ns* ÷ *b*) ÷ “floor” (4074 ÷ (*aos* + 61)) x 1.1
4K object table index 1 (OBJT04X1)	*ns* ÷ (68 x *b*)
32K object table (OSMOTS32)	***n*l ÷ ((32 746 ÷ (*aos* + 63)) x *cb*) x 1.1
32K object table index 1 (OBJT32X1)	nl ÷ (65 x *b*)

Note:

*Where “floor” means round to the next smaller integer before dividing by *ns* ÷ *b*. All objects stored in this table must be less than or equal to 3980 bytes in length. For example:

“floor” =(4074 ÷ (1500 + 61)) = 2.6 (or = 2 when rounded to down to the next lowest integer)

example: (34 100 ÷ 12) ÷ 2 = 14 209 x 1.1 = 15 630

**If your average object size (*aos*) is less than 32 746 bytes in length, the value (*aos* + 63) must be rounded to the next higher multiple of 4 kilobytes before using it in a formula; for example, 5 kilobytes rounds up to 8 kilobytes, 13 kilobytes rounds up to 16 kilobytes, and so on. Find the quotient of 32 746 ÷ (*aos* + 63) first. If the quotient is greater than 1, ignore any fractional remainder; if the quotient is less than 1, use the remainder as the result. Multiply the result by the value for *cb*, which yields the value of the divisor for the formula. Drop any fractional remainder from the divisor before dividing into the dividend *nl*. This quotient must be rounded to the next higher integer before being increased by the multiplier. The final result must be rounded to the next higher integer as well.

Optical Configuration Database

The optical configuration database (CBROAM) defines the optical hardware configuration and all of the optical volumes. It is a DB2 database and consists of the following tables:

Library
Contains one row for each optical library. The DB2 name of this table is **OLIBRARY**. There is a unique index on the library name.

Drive
Contains one row for each optical drive, whether stand-alone or library-resident. The DB2 name of this table is **DRIVE**. There are two indexes defined on the table; one is unique and one is not.

Slot
Contains one row for each of the slots in a 9246 optical library. The DB2 name of this table is **SLOT**. There is a unique index defined on the slot name in combination with the library name.

Volume Contains one row for each optical disk volume. The DB2 name of this table is **VOLUME**. There is a unique index on the volume serial number.

Deleted-Objects
Contains one row for each object waiting to be deleted from 3995 rewritable optical media. The DB2 name of this table is **DELOBJT**. There are two indexes defined on the table; one is unique and one is not.

Tape Volume Contains one row for each tape volume used by OAM for object storage. The DB2 name of this table is **TAPEVOL**. There is a unique index on the volume serial number.

Table 10 provides DASD space recommendations for storage of the CBROAM tables.

Table 10. CBROAM Space Recommendations

Description	DB2 Name	Primary Space 3390 Tracks	Secondary Space 3390 Tracks
Library Table	OCLIBTSP LNAMINDX	1 1	1 1
Drive Table	OCDRVTSP DNAMINDX DRIDINDX	5 1 1	1 1 1
Slot Table	OCSLTTSP SLIBINDX	1 1	1 1
Volume Table	OCVOLTSP VSERINDX	20 2	10 1
Deleted-Objects Table	OCDELTSP DVOLINDX DELOINDX	100 10 100	10 5 10
Tape Volume Table	OCTVLTSP TVOLINDX	10 2	1 1

These recommendations allow:
- 216 optical library definitions
- 1 320 drive definitions
- 1 404 slot definitions for as many as twenty-two 9246 library definitions
- 5 040 volume definitions
- 39 600 objects waiting for deletion from rewritable media
- 5 280 tape volumes to be used for storing objects

If your installation requires more entry space or if you are using direct-access storage that allows fewer than twelve pages per track, the above space recommendations may need to be increased.

OAM Administration Database

An additional database is needed for object management. This database is identified by the data set name qualifier OAMADMIN. Although specific calculations could be made for exact tracks needed based on the number of storage class names, management class names, and collection names used by your installation, experience has shown that the recommendations in Table 11 on page 62 should be adequate.

Table 11. OAMADMIN Space Recommendations

Description	Data Set Qualifier	Primary Space 3390 Tracks	Secondary Space 3390 Tracks
Management Class ID Table	MCIND CBRMGTX CBRMGTY	1 1 1	1 1 1
Storage Class ID Table	SCIND CBRSTOX CBRSTOY	1 1 1	1 1 1
Collection Name Table	COLIND CBRCLTX1 CBRCLTX2 CBRCLTX3	2 1 1 1	2 1 1 1

These recommendations allow the maximum 32,767 storage class names, the maximum 32,767 management class names and over 60,000 collection names.

Object Storage on Removable Media

If your installation is going to use optical, tape, or both types of storage, you must estimate the amount of optical disk and tape cartridge storage that will be needed to support your OAM implementation. Factors to consider should include the number of:

- Optical libraries required per day
- Optical disks, tape cartridges, or both required per year
- Shelf-resident optical disks, tape cartridges, or both
- Optical, tape, or both types of libraries
- Stand-alone or operator-accessible optical disk drives
- Tape stand-alone drives, ATLDSs, and MTLs, or a combination of these devices

This section provides techniques for calculating optical and tape space requirements based on these considerations. See “Optical Resources” on page 74 for an example of how these computations are applied.

Constants

Use the information in Table 3 on page 25 and Table 4 on page 37 regarding storage slot and cartridge capacities as constants for DASD resource calculations for your optical and tape configurations.

Formulas

The following formulas can be used to calculate storage (optical, tape, or both) estimates.

Note: The term “cartridge” in the following formulas refers to optical disk cartridges and tape cartridges. If you are only using one type of medium within your installation, simply calculate according to the needs of your storage management policy. If you are using both optical and tape storage, consider cartridges needed for both media when calculating the formulas in Table 12 on page 63.

Table 12. Formulas for Calculating Storage Requirements

Consideration Factor	Calculation
Cartridges per Day	# cartridges required per day equals: # megabytes written per day ÷ # of megabytes per cartridge *where*: # of megabytes written per day equals: objects created per day *x* object size in megabytes.
Cartridges per Year	# cartridges required per year equals: total # of cartridges per day *x* workdays per year
Shelf-Resident Cartridges	# shelf-resident cartridges equals: # cartridges required per year *x* retention period in years
Storage Group Adjustment	storage group adjustment equals: # of storage groups *x* # of active drives Therefore, total number of cartridges required per day equals: # cartridges required per day + storage group adjustment
Libraries for library-resident cartridges	# libraries required to hold library-resident cartridges equals: # of days library resident *x* (# cartridges required per day ÷ # slots in the library)
Note: # indicates a total number *x* indicates multiplication ÷ indicates division + indicates addition	

Cartridges per Day

The number of objects created per day should include only those objects stored on optical, tape, or both media types. If several objects of different sizes will be written to optical, tape, or both media types, calculate the number of megabytes written per day for each object size and sum the results to get the total number of megabytes written per day. See Table 12 for more information on this calculation.

Attention: You need to determine what value should be assigned to the time periods within your calculations. For example, the term *day* could be a calendar day or a workday depending on the requirements of your business. A week can be either a seven-day calendar week or a five-day workweek (or in some environments, this time frame can even be less). A year might include all the days of the year (including weekends and holidays) or it might only include the regular workdays for your installation. Include these factors in your calculations. Remember that OSMC does not recognize the difference between workdays, weekends, or holidays. Take this factor into account in your calculations for resource planning.

Each cartridge can contain objects from only one Object storage group; therefore, on the first day that optical, tape, or both types of storage are used, you will need at least one cartridge for each Object storage group. If you plan to create backup copies of objects, remember to calculate the number of cartridges needed for each Object Backup storage group (use the *cartridges per day* formula in Table 12). On the first day that your installation creates backup copies, you will need at least one cartridge of the appropriate media type for each Object Backup storage group.

Adjusting for Storage Groups and Active Drives

If multiple drives will be used for writing objects from one or more Object storage groups, increase the number of cartridges required per day by following the calculation for *storage group adjustment* in Table 12.

Effectively Using Optical Volume Space

The usage of optical volume space is affected by the size of the objects and how the objects are written: chained by the storage management cycle or unchained by direct write to optical media. Table 13 provides an estimate of optical volume usage for media used with a 9246 optical drive. For 3995 capacity information, refer to *3995 Introduction and Planning Guide*.

Table 13. Effective Optical Volume Usage for IBM 9247 Media

Object Size in KB	Effective Utilization	
	Data Written by Storage Management Cycle (Chained)	**Data Written Directly to Optical Volume (Unchained)**
40 000	100%	93%
20 000	100%	47%
10 000	100%	23%
5 000	100%	12%
4 000	100%	9%
3 000	84%	7%
2 000	56%	5%
1 000	28%	2%

Attention: In subsequent calculations, ensure that you use a value for the total number of cartridges *required-per-day* that is large enough to reflect your planned use of storage groups, multiple drives, and direct write to optical and or tape. The validity of those calculations will depend on the accuracy of your total number of cartridges *required-per-day* estimate.

Cartridges per Year

Use the calculations in Table 12 on page 63 for *cartridges-per-year* to determine the number of cartridges needed to satisfy your yearly medium requirements for your storage environment.

Shelf-Resident Cartridges

Use the calculations in Table 12 on page 63 for *shelf-resident cartridges* to determine the number of shelf-resident cartridges needed to satisfy your yearly medium requirements for your stand-alone and pseudo library environments.

Determining Library Requirements

The number of libraries required for an OAM implementation is influenced by several factors:

- Number of libraries required to hold library-resident cartridges
- Number of libraries required to satisfy the maximum retrieval rate of objects on optical, tape, or both types of storage
- Number of libraries required for the storage management cycle to complete within the allotted processing period.

The largest of the three numbers represents the actual number of libraries that you should plan to install. The following guidelines concerning libraries can help you evaluate your library needs.

Libraries for Library-Resident Cartridges

To ensure you correctly estimate the appropriate amount of libraries (optical, tape, or both) to hold all of your library-resident cartridges, refer to the calculation under *libraries for library-resident cartridges* in Table 12 on page 63 to assist you.

Libraries for Maximum Retrieval Rate

As a rule, each 3995 optical library can handle up to 200 mounts per hour and still provide an acceptable response time. If mount activity exceeds this rate, you may experience long delays on retrieval because of queued requests. To correct this situation, consider either installing an additional optical library or keeping more objects on DASD or tape.

Libraries for Storage Management Cycle Processing

Your installation should include in its regular schedule a period of time during which the storage management cycle can run. For example, you might execute the storage management cycle every day during third shift. During this time, OAM moves objects between optical disk volumes, tape volumes, and DASD. If this processing period is short, it may be necessary to install additional libraries to prevent contention caused by the following situations:

- Several Object storage groups are processed concurrently (controlled by the MAXS parameter).
- Multiple drives are used concurrently for a given Object storage group.
  - Optical drive usage is controlled by the DRIVE STARTUP THRESHOLD storage group parameter in ISMF
  - Tape drive usage is controlled by the use of the TAPEDRIVESTARTUP (threshold in MB) keyword on the SETOAM statement for each Object storage group

For information concerning effective object utilization for IBM 3995 media, refer to the *3995 Introduction and Planning Guide*.

XCF Resource Estimation

In an effort to best use the resources of the cross-system coupling facility (XCF), you should first use the default transport classes and run RMF reports with XCF usage to determine if customization is needed. See "Using Appropriate Transport Classes within XCF" on page 179 and refer to *RMF User's Guide* for more information. If you try to establish a configuration where the hardware is on the same system where the highest needs are for that library (storage group level, OSMC processing level, or user grouping), the cross-system overhead is reduced. OAM processing in an OAMplex will increase XCF resource overhead with small messages used to communicate changes in the configuration during normal processing, and larger messages for object reads or writes that require cross-system processing.

Concluding Resource Estimation

After your resource estimations have been calculated, compare those projections with the resources you have available. Determine the additional resources you will need to support your planned use of OAM, and develop a schedule for obtaining those resources. You can order some items prior to OAM installation; other items may not be needed until later, as your use of objects increases. Make sure your project plan includes time to order, install, and test essential resources before OAM is installed.

Preparing the Physical Environment

OAM itself does not require changes to the physical environment; however, if you will be using optical storage subsystems or tape library subsystems for the first time, you can prepare for their installation. For detailed information concerning optical storage subsystems, refer to the *RPQ Optical Storage Subsystem Product User's Manual* and to the *LAN Channel Station Installation and Test* or to the *3995 Introduction and Planning Guide*. For information regarding the tape library dataservers, refer to *TotalStorage Automated Tape Library (3495) Installation Planning and Migration Guide*, *TotalStorage Automated Tape Library (3494) Introduction and Planning Guide* and the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Preparing for Installation and Customization

Once the conceptual groundwork for OAM has been completed, you are ready to install OAM and to translate the theoretical model into a functioning system. The basic procedure for installing OAM consists of the following steps:

1. Ensure that all hardware and software prerequisites have been met.
2. Prepare the processing environment to interface with OAM. This entails the following tasks:
   a. Change DB2 installation parameters.
   b. Add the DB2 transport classes to the coupling facility resource management (CFRM) policy if you are setting up an OAMplex with DB2 data sharing.
   c. Change CICS installation parameters.
   d. Change non-CICS installations (Batch, TSO, IMS).
   e. Modify the system libraries.
   f. Verify DB2 installation.
   g. Create DB2 databases and application plans.
   h. Define user catalogs.
   i. Define the PARMLIB CBROAM*xx* member if you are using object tape support, setting up an OAMplex, or customizing optical support, or customizing OSMC support. Put valid SETOAM, SETOPT, SETOSMC, or OAMXCF statements into the created member as appropriate. Add the OAM=*xx* specification to the OAM cataloged procedure.
3. Specify the SMS definitions and programs that are used by OAM. To accomplish this, perform the following tasks:
   - Translate the business analysis into technical terms.
   - Create OAM definitions with ISMF.
4. Validate the installation.
5. Activate the configuration.
6. Use the Installation Verification Program to verify the success of the object support installation.

After installation is complete, you can customize OAM.

To simplify the installation process, a library of sample jobs and other useful data sets (SAMPLIB) is shipped with the product. "Appendix B. Sample Library Members" on page 381 contains listings or prologs of many of the SAMPLIB members.

Planning to Program Applications for OAM

The *z/OS DFSMS OAM Application Programmer's Reference* manual describes the OSREQ macro, the programming interface provided by OAM. Refer to this document for detailed information about programming applications which use OAM.

As you consider OAM programming applications, keep in mind the following items:

- Application design plays a significant role in OAM performance. For example, careful use of buffering can reduce virtual storage requirements.
- Application programs are responsible for synchronizing OAM-related DB2 databases (for example, using SYNCPOINT under CICS).
- The auto-delete installation exit can be programmed specifically for an application, as described in "Auto-Delete Installation Exit (CBRHADUX)" on page 509.
- ACS routines must be programmed.
- Maintenance of the auto-delete installation exit and ACS routines is generally the responsibility of the storage administration team; however, a particular application may require modification to use the exit and the ACS routines.

Planning to Administer OAM

Product implementation only *begins* with installation; your planning must also include preparation for ongoing administration of the product. The focal point of OAM administration is the storage administration team, which is responsible for the following tasks:

- Monitoring and maintaining the SMS configuration through ISMF
- Monitoring and maintaining DB2 databases
- Tuning OAM
- Establishing recovery procedures
- Destroying expired data

"Chapter 4. Administering OAM" on page 151, provides tools and techniques for performing these functions.

The success of an OAM implementation depends significantly on the quality of the support staff. The storage administration team should receive intensive training before OAM installation and encouragement to keep abreast of current technology through continuing education. Your IBM marketing representative can help you identify appropriate publications and training opportunities.

Preparing to Operate OAM

Daily operation of the OAM system is the responsibility of the operations staff. Operator tasks are explained in "Chapter 5. Operating OAM and OTIS Address Spaces and OSMC Functions" on page 217. For information concerning operator tasks in reference to the tape library dataservers, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

To ensure successful OAM operation, the operations staff should work closely with the storage administration team and the application team to coordinate support efforts. Be sure to update your installation's operating procedures manual to include OAM-related tasks.

Operators need to have in-depth knowledge about the hardware used by OAM, especially optical and tape storage devices. They should be encouraged to keep abreast of current technology through continuing education. Your IBM marketing representative can help you identify appropriate publications and training opportunities.

OAM Planning Case Study

A large company is in the process of planning for OAM implementation. This case study documents part of their planning effort.

Object Characterization

During the business analysis phase, the planning team members determined that they will be processing two types of objects, which they refer to as summary objects and detail objects. Summary objects are small; detail objects are considerably larger. On an average workday, about 10 000 summary objects and 10 000 detail objects are created.

Establishing Performance Objectives and Availability Requirements

The planning team analyzed data usage patterns to establish performance objectives and availability requirements for their two object types. Summary objects are used for 30 calendar days, and response time must be fast during that period; however, at the end of 30 calendar days, the objects are no longer used and may be deleted. Detail objects have a more complex life cycle. During the first seven calendar days after creation, detail objects are accessed frequently, and fast response is needed. After that time, retrieval frequency drops, as does the response-time requirement. Detail objects are rarely used after 180 calendar days, but the company is required by law to keep them on file for five years (1 825 calendar days).

Table 14 summarizes the characteristics of the company's objects.

Table 14. Object Characteristics

Characteristics	Detail Objects	Summary Objects
Object size in bytes	64 000	3 000
Number created per workday	10 000	10 000
Number of workdays that fast retrieval is required (less than 1-second response time)	7	30
Number of calendar days that medium retrieval is required (less than 20-second response time)	180	—
Number of calendar days from creation after which the object is rarely accessed	180	—
Number of calendar days from creation after which the object can be deleted	1 825 (5 years)	30
Maximum retrieval rate required (objects per hour) for fast retrieval	1 000	2 000
Maximum retrieval rate required (objects per hour) for medium retrieval	100	—
Maximum retrieval rate required (objects per hour) for slow retrieval	10	—
Number of backup copies required	1	0

Establishing Collections and Constructs

The planning team established two collections, one for each type of object. They developed storage classes (SC) to reflect the three service levels indicated by the business analysis, and they created management classes (MC) to correspond to the transition points in the objects' life cycles (that is, when they expire or when their performance objectives change). Then they determined which storage and management classes would be the default initial class assignments for each collection. Finally, specifications were drafted for the ACS routines that control an object's storage group, storage class, and management class assignments.

Also, during the resource estimation phase, the planning team concluded that the 10 000 detailed objects that are being stored daily through the workweek must be backed up for disaster recovery purposes. The data contained within the detailed objects is critical business data that must be recovered in the event of a disaster.

SC Name — **Description**

FASTPERF
SC for objects with high-performance requirements (less than 1-second response time preferred). This storage class was designed to be the default initial storage class assignment for both summary and detail objects.

MEDPERF
SC for objects with medium-performance requirements (less than 20-second response time preferred). This storage class was designed primarily for detail objects for which high performance is no longer necessary.

LOWPERF
SC for objects with low-performance requirements (more than 20-second response time acceptable). This storage class was designed primarily for detail objects that are rarely retrieved.

MC Name — **Description**

EXP30
Thirty calendar days from the date of creation, objects with this management class should be processed by the storage management cycle. The expiration attributes in the definition for this management class will indicate that objects in this class can be deleted after 30 calendar days. EXP30 was developed primarily for summary objects and is their default initial management class assignment.

TRAN7
Seven calendar days from the date of creation, objects with this management class should be processed by the storage management cycle. This class was developed primarily for detail objects. After seven calendar days, the performance objective for detail objects changes from fast to medium, and a new storage class assignment is needed. TRAN7 is the default initial management class assignment for detail objects.

TRAN180
On the 180th calendar day from the date of creation, objects with this management class should be processed by the storage management cycle. This class was developed primarily for detail objects. After 180 calendar days, the performance objective for detail objects changes from medium to low and a new storage class assignment is needed. TRAN180 should be assigned to detail objects that have a storage class assignment of MEDPERF.

EXP1825
Five years (1825 calendar days) from the date of creation, objects with this management class should be processed by the storage

management cycle. The expiration attributes in the definition for this management class will indicate that objects in this class can be deleted after five years. EXP1825 was developed for detail objects and should be assigned to all detail objects that have a storage class assignment of LOWPERF.

Note: Management class definitions support parameters other than those shown in this case study (for example, backup requirements and retention period for an object since last use). Review all of the parameters before you attempt to classify the objects in your applications.

Coding ACS Routines to Implement Class Transitions

The company's technical staff was able to code ACS routines that use these classes, along with the installation's object-naming conventions, to implement class transitions. During every storage management cycle, OSMC checks each object that is scheduled for storage management processing to see if it is due for expiration. If it is, OSMC deletes it; otherwise, the management class assignment is used to determine if class transition is needed. For class transition, the ACS routines are invoked.

The general logic of the storage management cycle and the ACS routines is as follows (for objects created on day *X*):

1. For definitions of collections:
   - If the collection name is detail and the object name is Null, then SC=FASTPERF and MC=TRAN7.
   - If the collection name is summary and the object name is Null, then SC=FASTPERF and MC=EXP30.

   These default values are stored in collection name entry in the ICF catalog when the first object is stored to that collection. When neither storage class nor management class is specified on a request to store an object into one of these collections, the object is assigned the default classes associated with that collection.
2. At *X* + 7 calendar days, objects in MC=TRAN7 are processed by the storage management cycle. MC=TRAN7 does not specify that these objects should be deleted, so the ACS routines are invoked. If an object's name begins with *D* (the naming convention for detail objects) and has SC=FASTPERF, the ACS routines change the class assignments to SC=MEDPERF and MC=TRAN180. This change in storage class can cause the object to be relocated in the object storage hierarchy. For example, the object might move from DASD to optical disk to a tape volume, or any combination therein.
3. At *X* + 30 calendar days, objects in MC=EXP30 are processed by the storage management cycle. The expiration attributes in the definition for MC=EXP30 specify that objects in this class should be deleted at *X* + 30 calendar days, so that the objects are deleted by OSMC.
4. At *X* + 180 calendar days, objects in MC=TRAN180 are processed by the storage management cycle. MC=TRAN180 does not specify that these objects should be deleted, so that the ACS routines are invoked. If an object's name begins with *D* (the naming convention for detail objects) and has SC=MEDPERF, the ACS routines change the class assignments to SC=LOWPERF and MC=EXP1825. The change in storage class will be recorded in the object directory.

   Attention: If the performance characteristics of the storage class are intended to be physically realized in shelf storage, movement to shelf does not occur in object transition processing unless the object is located on DASD at the time that transition processing begins and a pseudo library is defined for the storage group. In such a case, where the object was located on DASD prior to transition, location on a shelf in a pseudo library can only be guaranteed in cases where the only libraries defined in the storage group are pseudo libraries.

You can conduct an analysis using DB2, SPUFI, or QMF™ to determine which volumes in the configuration presently located within real libraries contain only objects with a storage class intended for the shelf or pseudo library. Those volumes found can be ejected from the library and placed in their assigned shelf location. The DB2 analysis will query the object directory table for each storage group required. Query the storage class table to determine the storage class identifier with which to qualify queries.

5. At *X* + 1825 calendar days, objects in MC=EXP1825 are processed by the storage management cycle. The expiration attributes in the definition for MC=EXP1825 specify that objects in this class should be deleted at *X* + 1825 calendar days, so that the objects are deleted by OSMC.

Resource Estimation

Having completed the business analysis phase, the planning team used their object size and activity estimates to evaluate the DASD, tape, and optical storage resources that would be needed to support OAM.

DASD Resources

During the resource estimation phase, the planning team used the formulas on page “Variables” on page 59 to determine their DASD storage needs.

Calculating DASD Storage for an Object Storage Database

The following example calculates the direct-access storage needed for one object storage database. These calculations would be repeated for each object storage database within OAM. The specific values for the example calculations are based on the following conditions:

- During each workday 10 000 objects, each 3 000 bytes long, are stored in database GROUP00. A second case is also shown in the example calculation for objects averaging 1 500 bytes long.
- During each workday 10 000 objects, each 64 000 bytes long, are stored in database GROUP00. A second case is also shown in the example calculation for objects averaging 9 000 bytes long.
- The 3 000-byte objects that have been in the database for 30 calendar days are deleted from the database.
- The 64 000-byte objects that have been in the database for seven calendar days are moved from the database to optical storage, where they will remain for five years (1 825 calendar days) before being deleted.
- New objects are stored in the database before any existing objects are deleted or moved.
- To allow for new objects exceeding the predicted maximum number, an extra 10% space contingency is added.
- The database for object storage resides on a 3390-type, direct-access storage device.

Calculating the Number of Objects Resident on DASD, Optical, and Tape

The planning team began by calculating the number of objects that will be resident on direct-access storage, tape, and optical storage.

1. The maximum number of small objects resident on direct-access storage is determined as the number of summary objects created daily (10 000) plus the number of summary objects already resident in the database (10 000 x 30 days) plus the 10% contingency. The maximum number of direct-access-resident small objects is identified in the formulas as the variable *ns*.

   Calculate the value of *ns* as:

   ns = (10 000 + (10 000 x 30)) x 1.1 = 341 000
2. The maximum number of large objects resident on direct-access storage is determined as the number of detail objects created daily (10 000) plus the number of detail objects already resident in the database (10 000 x 7 days) + the 10% contingency. The maximum number of direct-access-resident large objects is identified in the formulas as the variable *nl*.

   Calculate the value of *nl* as:

   nl = (10 000 + (10 000 x 7)) x 1.1 = 88 000
3. The number of objects that are stored within this Object storage group, but not within the small or large object tables, is determined as the number of summary objects retained within this Object storage group minus the number of summary objects on direct-access storage, plus the number of detail objects retained within this Object storage group, minus the number of detail objects on direct-access storage. Neither calculation can have a result less than zero.
   a. The number of summary objects retained within this Object storage group but not stored in the small object table is calculated as the number of summary objects stored each day, times the maximum number of days the objects are retained, minus the maximum number of direct-access resident summary objects (value of variable *ns* above).

      Calculate the number of small objects on optical and tape storage as:

      os = ((10 000 x 30) x 1.1) − *ns* = 330 000 - 341 000 = 0
   b. The number of detail objects retained within this Object storage group but not stored in the large object table is calculated as the number of detail objects stored each day, times the maximum number of days the objects are retained, minus the maximum number of direct-access resident detail objects (value of variable *nl* above).

      Calculate the number of large objects on tape and optical storage as:

      ol = ((10 000 x 1307) x 1.1) − *nl* = 14 377 000 − 88 000 = 14 289 000
   c. The total number of objects retained on tape and optical storage on any given day is the sum of the number of summary objects on tape and optical (*os* above) plus the number of detail objects on tape and optical (*ol* above).

      Calculate the total number of objects retained on tape and optical storage as:

      ot = *os* + *ol* = 0 + 14 289 000 = 14 289 000
4. The total number of objects that need to be referenced in the GROUP00 database is the sum of the maximum number of direct-access-resident small objects (*ns* above), plus the maximum number of direct-access-resident large objects (*nl* above), plus the total number of objects retained on tape and optical (*ot* above). This value is used in the formulas as the variable *nt*.

   Calculate the value of *nt* as:

   nt = *ns* + *nl* + *ot*

   nt = 341 000 + 88 000 + 14 289 000 = 14 718 000

Now calculate the storage needed for the object storage database data sets (see Table 15 on page 73).

Table 15. Storage Calculations for the Object Storage Database Data Sets

TABLE, INDEX, or DATA SET	CALCULATIONS
Object directory data set **GROUP00.OSMDTS**	tracks = *nt* ÷ (35 x *b*) example: 14 718 000 ÷ (35 x 12) = 35 043 cylinders = 35 043 ÷15 = 2336 **This results in 35 043 tracks or 2336 cylinders**
Object directory index 1 **GROUP00.OBJDIRX1**	tracks = *nt* ÷ (260 x *b*) example: 14 718 000 ÷ (260 x 12) = 4717 cylinders = 4717 ÷15 = 315 **This results in 4717 tracks or 315 cylinders**
Object directory index 2 **GROUP00.OBJDIRX2**	tracks = *nt* ÷ (166 x *b*) example: 14 718 000 ÷ (166 x 12) = 7389 cylinders = 7389 ÷15 = 493 **This results in 7389 tracks or 493 cylinders**
Object directory index 3 **GROUP00.OBJDIRX3**	tracks = *nt* ÷ (68 x *b*) example: 147 180 000 ÷ (68 x 12) = 18 037 cylinders = 18 037 ÷15 = 1203 **This results in 18 037 tracks or 1203 cylinders**
Small object table index **GROUP00.OBJT04X1**	tracks = *ns* ÷ (68 x *b*) example: 341 000 ÷ (68 x 12) = 418 cylinders = 418 ÷15 = 28 **This results in 418 tracks or 28 cylinders**
Small object table (object size 3000 bytes) **GROUP00.OSMOTS04**	tracks = (*ns* ÷ *b*) x 1.1 example: (341 000 ÷ 12) x 1.1 = 31 259 cylinders = 31 259 ÷15 = 2084 **This results in 31 259 tracks or 2084 cylinders**
Small object table (object size 1500 bytes) **GROUP00.OSMOTS04**	tracks = (*ns* ÷ *b*) ÷ “floor” (4074 ÷ (*aos* + 61)) x 1.1 “floor” = (4074 ÷ (1500 + 61)) = 2.6 (= 2 when rounded to down to the next lowest integer) example: (34 100 ÷ 12) ÷ 2 = 14 209 x 1.1 = 15 630 cylinders = 15 630 ÷15 = 1042 **This results in 15 630 tracks or 1042 cylinders**
Large object table index **GROUP00.OBJT32X1**	tracks = *nl* ÷ (65 x *b*) example: 88 000 ÷ (65 x 12) = 112 cylinders = 112 ÷15 = 8 **This results in 112 tracks or 8 cylinders**
Large object table **GROUP00.OSMOTS32**	cylinders = *nl* ÷ (32 746 ÷ (*aos* + 63) x 22) x 1.1 first calculating: (32 746 ÷ (64 000 + 63)) = approx. 0.511 example: 88 000 ÷ (0.511 x 22)= 7828 x 1.1 = 8611 **This results in 8611 cylinders**
Large object table (object size 9000 bytes) **GROUP00.OSMOTS32**	(32 746 ÷ (*aos* + 63)) results in: 32 746 ÷ 12 = 2.73 (= 2 when rounded down to the next lowest integer) example: 88 000 ÷ (2 x 22) = 2000 x 1.1 = 2200 **This results in 2200 cylinders**

Table 16 summarizes the storage space calculations.

Table 16. Sample Storage Space Calculation Results

Database Name Qualifiers	Tracks Needed	Cylinders Needed
GROUP00.OSMDTS	35 043	2336
GROUP00.OBJDIRX1	4717	315
GROUP00.OBJDIRX2	7389	493
GROUP00.OBJDIRX3	18 037	1203
GROUP00.OBJT04X1	418	28
GROUP00.OSMOTS04 (object size 3000 bytes)	31 259	2084
GROUP00.OSMOTS04 (object size 1500 bytes)	15 630	1042
GROUP00.OBJT32X1	112	8
GROUP00.OSMOTS32	—	8611
GROUP00.OSMOTS32 (object size 9000 bytes)	—	2200

It is important to remember that the object directory table has an entry for *every* object within that Object storage group. This directory includes objects in the 4K object table and in the 32K object table, plus objects on tape and optical storage. The 4K object table and the 32K object table have only the objects that are resident on direct-access storage. When an object is stored directly on tape or optical storage, an entry is created in the object directory, but the object itself is not stored in either the 4K or the 32K object tables. For all objects stored directly on tape or optical storage, the calculations for small and large objects on tape or optical storage (see item 3 on page 72) will provide the values needed for the object directory space.

Optical Resources

During the resource estimation phase, the planning team used the formulas on page 62 to determine their optical storage needs. They used the object size and volume estimates that were developed during the business analysis phase (see Table 14 on page 68). The following assumptions were used for the calculation:

- Only the 10 000 detail objects would eventually be written to optical disk.
- The optical residence period is 180 calendar days from creation date minus the seven calendar days on DASD.

Table 17 on page 75 represents the optical resources calculated by the planning team for each of the media types.

| **Note:** The free space available on double-density, quad-density, and 8x-density WORM platters might not match the formatted capacity. The hardware holds a certain percentage of sectors in reserve based on the media type.

Table 17. Optical Resource Requirements for Library Resident Data

MEDIA	12-inch LMSI*	SD WORM**	SD REWR**	DD WORM**	DD REWR**	QD WORM**	QD REWR**	8X WORM**	8X REWR**
MB of data to be written per workday	625	625	625	625	625	625	625	625	625
Cartridge capacity for user data	1950 MB	590 MB	580 MB	1180 MB	1160 MB	2520 MB	2394 MB	**4592 MB ***5001 MB	**4335 MB ***4722 MB
# of disks to be used per workday	0.321	1.059	1.078	0.530	0.539	0.248	0.261	**0.136 ***0.125	**0.144 ***0.132
# of calendar days the data is needed to be library resident	173	173	173	173	173	173	173	173	173
Total # of cartridges needed to retain library resident data	56	184	187	92	94	43	45	**24 ***22	**25 ***23
# of slots in the library	64	144	144	144	144	52 104 156 258	52 104 156 258	52 104 156 258	52 104 156 258
# of libraries needed to hold library resident data	1	2	2	1	1	1	1	1	1

Note:
MB = megabyte (1 048 576 bytes)
GB = gigabyte (1 073 741 824) bytes
SD = single-density
DD = double-density
QD = quad-density
8x = 8x-density
REWR = rewritable media
WORM = write-once, read-many
DD/QD WORM also includes continuous composite WORM (CCW) media.

*Media for 9246 optical library.

**For 1024 sector media.

***For 2048 sector media.

Estimating the Number of Libraries Required for Maximum Retrieval Rate

The next factor to consider in estimating libraries is the number of libraries needed to satisfy the maximum retrieval rate for objects on optical storage. The application work load (as estimated in Table 14 on page 68) has a maximum retrieval rate of 100 objects per hour. Even if each retrieval results in mounting a different disk, one library is sufficient to satisfy this requirement. If you are making backup copies of objects, you need to plan for additional optical disks to accommodate them. The megabytes of data written per day is related to objects assigned to a management class that specifies an auto backup value of *yes*.

Tape Resources

To estimate the number of tape cartridges needed per workday, the planning team determines the amount of data to be backed up per workday. The amount of OAM data being backed up per workday is 572.2 megabytes (10 000 objects per day x 60 000 bytes per object = 600 000 000 bytes, which equates to 585 938 kilobytes, or 572.2 megabytes).

The number of tape cartridges needed per workday is determined by the following factors:

- The amount of data written to tape per workday
- The type of tape cartridge (cartridge system tape or enhanced capacity cartridge system tape media)
- The type of tape cartridge (cartridge system tape, enhanced capacity cartridge system tape, or high performance media)
- The recording technology used on the tape cartridges:
  - 18-track format written on IBM 3480 or 3490 base models
  - 36-track format written on IBM 3490E enhanced capability models
  - 128-track format written on IBM 3590 models
  - 256-track format written on IBM 3590-E models
- The number of tape drives and tape volumes available on each system within an OAMplex, if writing backups to tape in a Parallel Sysplex environment. Additionally, the take into consideration the number of tape volumes needed if transitioning storage groups to tape.

Assuming the tape cartridges are filled to 100% of their estimated capacity, Table 18 shows the number of tape cartridges of each type of recording format combination that would be needed daily.

Table 18. The Number of Tape Cartridges Needed Per Workday

Media Type	**MB Written To Tape Per Workday**	**Cartridge Capacity For User Data**	**# Tape Cartridges Used Per Workday**
Cartridge System Tape (18-Track Format)	572.2	200 MB	2.86
Cartridge System Tape (36-Track Format)	572.2	400 MB	1.43
Enhanced Capacity Cartridge System Tape (36-Track Format)	572.2	800 MB	.715
IBM High Performance Cartridge Tape (128-Track Format)	572.2	10 GB	.057
IBM Extended High Performance Cartridge Tape (128-Track Format)	572.2	20 GB	.029
IBM High Performance Cartridge Tape (256-Track Format)	572.2	20 GB	.029
IBM Extended High Performance Cartridge Tape (256-Track Format)	572.2	40 GB	.014
Note: MB = 1 048 576 bytes GB = 1 073 741 824 bytes			

Chapter 3. Migrating, Installing, and Customizing OAM

This chapter explains how to install OAM, customize it for your business and processing environments, and verify that the installation is complete and correct. The basic procedure for installing OAM consists of the following topics:
- “Verifying Hardware and Software Prerequisites” on page 77
- “Preparing the Processing Environment” on page 77
- “Installation and Migration Checklist” on page 79
  - “Specifying the SMS Definitions and Programs Used by OAM” on page 128
  - “Validating and Activating the Configuration” on page 147
  - “Verifying Object Support Installation With IVP” on page 147
- “Moving OAM from One System to Another” on page 148

In addition to these basic installation requirements, this chapter also includes a description of an optional auto-delete installation exit that can be used to further customize your installation.

To simplify the installation process, a library of sample jobs and other useful data sets (SAMPLIB) is shipped with the product. This chapter includes instructions for using SAMPLIB; “Appendix B. Sample Library Members” on page 381 contains listings of many of the SAMPLIB members. Before running any SAMPLIB job, remember to change the JCL to reflect your installation’s requirements (for example, accounting information and data set names). Refer to *z/OS MVS JCL Reference* for additional information.

| **Note:** Unless otherwise indicated, once you migrate to the current release, there
| are no necessary steps to migrate backward other than to run the BIND jobs.

Verifying Hardware and Software Prerequisites

Before installing OAM, you must first verify that the hardware and software requirements specified in “Hardware” on page 55 and “Software” on page 56 have been met. Ensure that all the prerequisites have been installed and thoroughly tested to verify that they operate correctly in your processing environment before proceeding with any other installation steps.

Preparing the Processing Environment

For OAM to communicate with the rest of the processing environment, system software must be altered to interface with OAM and may entail these tasks:
- Changing DB2 installation parameters and modifying the Computer Facility Resource Manager (CFRM) policy if necessary
- Changing CICS installation parameters
- Modifying the installation exit to handle deleted objects
- Changing system libraries
- Creating DB2 databases for object tables and directories
- Creating optical configuration databases
- Creating and binding DB2 packages
- Creating OSR application plans, or creating LCS, ISMF, and OSR application plans
- Creating OSMC application plans
- Verifying DB2 installation
- Defining User Catalogs
- IPLing the System

- Specifying the SMS definitions and programs used by OAM

Procedures for each of these tasks are presented in this chapter.

Preparing for Migration or Installation

Upon completion of any migration prerequisites, you should proceed with the following installation and migration checklist, paying particular attention to the guidance that is directed at installations that have had OAM installed previously.

Important: The term *migration* means to upgrade from one version of DFSMS to this current version. This assumes that you were previously using OAM for storing objects and that you will continue to use OAM for object storage upon installation of this current version or release. The term *installation* indicates that you have not installed OAM in any of the previous versions of DFSMS/MVS, OS/390, or z/OS, and are therefore installing OAM for the first time with this version of the product.

Table 19 assists you with your migration from a previous version of DFSMS/MVS or OS/390 to the current version. It describes which checklist steps must be performed, not performed, or verified, depending on which release or version you are migrating from. It also details the appropriate steps to perform if you are installing OAM for the first time. Use this information in conjunction with the detailed checklist steps that follow it.

Table 19. Previous Release Migration/Installation Checklist Performance Criteria

Checklist Steps	DFSMS 1.4.0	DFSMS 1.5.0	OS/390 V2R10	New Installation (z/OS V1R3)
Changing DB2 Installation Parameters	V	V	V	P
Changing Customer Information Control System (CICS) Installation Parameters	V	V	V	P
Modify the Installation Exit to Handle Deleted Objects	V	V	V	V
Changing System Libraries	P	P	P	P
Creating DB2 Databases for Object Tables and Directories	V	V	V	P
Creating OCDB: **Modify, then run:** **CBRSAMPL SAMPLIB** **CBRSM150 SAMPLIB** **CBRSMR13 SAMPLIB** **CBRSMB2 SAMPLIB**	 N P P V	 N N P V	 N N P N	 P N N N
Creating and Binding DB2 Packages	P	P	P	P
Creating OSR Application Plans*	P	P	P	P
Creating OSMC Application Plans	P	P	P	P
Creating LCS, ISMF, and OSR Application Plans	P	P	P	P
DB2 Installation Verification	P	P	P	P
Define User Catalogs	V	V	V	P
Initial Program Load (IPL) the System	P	P	P	P
Specify the Storage Management Subsystem (SMS) Definitions and Programs Used by OAM	V	V	V	P
Perform the OAM Installation Verification Program	P	P	P	P
*Perform only if NOT performing the Creating LCS, ISMF, and OSR Application Plans step. P = Perform N = Do not Perform V = Verify (perform if necessary)				

Installation and Migration Checklist

A number of steps are involved in the installation of OAM. A checklist to outline these steps and to assist in ensuring that all steps have been completed is provided. The detailed procedural steps follow this checklist. It is recommended that you read this entire section first so you know what to expect in this installation. As you actually perform the installation, reread the appropriate segment for each step in the checklist prior to performing it. Use this checklist only as a guide, as it does not contain the detailed information you need to perform each step correctly.

Before proceeding with this installation checklist, you should have verified the prerequisite hardware and software requirements (see “Hardware” on page 55 and “Software” on page 56 for more information).

If OAM has been previously installed, then you must carefully analyze the following steps for your particular installation. You will be given additional migration guidance for each step indicating that you need only review the step to ensure that you have completed it in your previous installation, or that you should perform the step regardless of whether you completed it in your previous installation, or that you should not perform the step. Remember, however, that each installation is unique, and you must carefully study these materials to ensure that you are taking the appropriate action for your installation environment.

“Changing DB2 Installation Parameters” on page 82

__ Step 1. Evaluate and select appropriate values.

“Changing CICS Installation Parameters” on page 84

__ Step 2. Update or create CICS PLT.

__ Step 3. Update CICS PPT.

__ Step 4. Update CICS SIT.

__ Step 5. Connect DB2 to CICS.

__ Step 6. Update DB2/CICS RCT.

__ Step 7. Copy CBRICONN to DFHRPL

Note: Perform steps 2 through 7 only if CICS is installed on your system.

“Modifying the Installation Exit to Handle Deleted Objects” on page 87

__ Step 8. Evaluate and implement auto-delete installation exit.

| **Note:** Perform this step **only** if you are running OSMC for expiration processing.

“Changing System Libraries” on page 87

__ Step 9. Update PARMLIB:

| __ a. Update IGDSMS*xx* PARMLIB member.

| **Note:** Do **not** perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.

| __ b. Update IEFSSN*xx* PARMLIB member.

| __ c. Update SMFPRM*xx* PARMLIB member.

| __ d. Update IEAICS*xx* PARMLIB member.

| __ e. Update CONSOL*xx* PARMLIB member.

__ Step 10. Create or update CBROAM*xx* PARMLIB members.

| **Note:** Do **not** perform this step if you are using DASD-only storage
| (no optical volumes or tape devices).

__ Step 11. Update PROCLIB:

__ a. Modify, if necessary, then run CBRIPROC SAMPLIB job.

| **Note:** Do **not** perform this step if you are using DASD-only
| storage (no optical volumes or tape devices) and you are
| not running OSMC for expiration processing.

__ b. Modify, if necessary, then run CBRAPROC SAMPLIB job.

| **Note:** Perform this step **only** if you start the OAM address
| space.

__ Step 12. Verify or create device numbers.

| **Note:** Do **not** perform this step if you are using DASD-only storage
| (no optical volumes or tape devices) and you are not running
| OSMC for expiration processing.

“Creating DB2 Databases for Object Tables and Directories” on page 116

__ Step 13. Add additional steps to the Database Creation Jobs, if necessary:
(CBRIALC0 and CBRISQL0)

__ Step 14. Modify the OAM Data Set Allocation Jobs:
(CBRIALC0, CBRIALCX, CBRIALCY)

Note: If DB2 data sets are being placed in an SMS storage group, you must properly prepare the environment (that is, ACS routine, and so on).

__ Step 15. Run the OAM Data Set Allocation Jobs:
(CBRIALC0, CBRIALCX, CBRIALCY)

__ Step 16. Modify the OAM Database Definition Jobs:
(CBRISQL0, CBRISQLX, CBRISQLY)

__ Step 17. Run the OAM Database Definition Jobs:
(CBRISQL0, CBRISQLX, CBRISQLY)

Note: Remember to modify the SAMPLIB jobs for your installation JOB card requirements and DB2 subsystem name and to start DB2 before running the SAMPLIB jobs.

“Creating the Optical Configuration Database” on page 120

| **Note:** Do **not** perform steps 18–21 if you are using DASD-only storage (no optical
| volumes or tape devices) and you are not running OSMC for expiration
| processing.

__ Step 18. Modify, if necessary, then run the CBRSAMPL SAMPLIB job (for first-time OAM installations).

__ Step 19. Run the CBRSMB2 SAMPLIB job.

Note: Do **not** perform this step at initial installation. Modify and run this step for migration purposes.

__ Step 20. Run the CBRSM150 SAMPLIB job.

Note: Do **not** perform this step at initial installation. Perform this step for migration purposes.

__ Step 21. Run the CBRSMR13 SAMPLIB job.

Note: After you run the CBRSMR13 job, you may need to run a DB2 reorganization after performing an ALTER to the table. Do **not** perform this step at initial installation. Perform this step for migration purposes only.

“Merging Object Tables and OCDB for an OAMplex” on page 121

__ Step 22. Run the CBRSMERG SAMPLIB job.

Note: Do **not** perform this step at initial installation. Modify and run this step for migration purposes. This job should only be run if you plan to merge multiple optical configuration databases (OCDB) into a single shared OCDB for an OAMplex.

__ Step 23. Run the CBRSG100 SAMPLIB job.

Note: Do **not** perform this step at initial installation. Perform this step only when all of the following conditions exist: you are setting up an OAMplex; you currently have multiple OAMs running on separate MVS images in a sysplex; and you want to merge two or more separate OAMADMIN tables, object storage databases, or both.

“Creating and Binding DB2 Packages” on page 122

__ Step 24. Run the CBRPBIND SAMPLIB job.

“OSR Application Plans” on page 122

__ Step 25. Run the CBRIBIND SAMPLIB job.

__ Step 26. Run the CBRIGRNT SAMPLIB job.

“OSMC Application Plans” on page 124

__ Step 27. Run the CBRHBIND SAMPLIB job.

Note: Do **not** perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.

__ Step 28. Run the CBRHGRNT SAMPLIB job.

Note: Do **not** perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.

“LCS, ISMF, and OSR Application Plans” on page 124

__ Step 29. Run the CBRABIND SAMPLIB job.

Note: Do **not** perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.

__ Step 30. Run the CBRAGRNT SAMPLIB job.

Note: Do **not** perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.

"Verifying DB2 Installation" on page 125

__ Step 31. Verify that all application plans have been created.

__ Step 32. Verify that all application plans have been authorized.

"Defining User Catalogs" on page 127

__ Step 33. Evaluate and implement user catalogs and policies.

"IPL the System" on page 128

__ Step 34. IPL the system.

"Specifying the SMS Definitions and Programs Used by OAM" on page 128

__ Step 35. Define the base SCDS.

__ Step 36. Define libraries and drives in the optical configuration database.

Note: You must perform this step at initial installation. During migration, you may optionally perform this step if you are adding or changing libraries or drives. Do **not** perform this step if you are using only DASD or tape storage (no optical volumes, drives, or libraries), or if you are running OSMC for expiration processing.

__ Step 37. Define storage groups (Object and Object Backup).

__ Step 38. Define storage classes.

__ Step 39. Define data classes.

__ Step 40. Define management classes.

__ Step 41. Define and test ACS routines.

__ Step 42. Validate and activate the configuration.

__ Step 43. Run the OAM IVP for Object Support.

The following procedural steps provide details to assist you in the performance of the checklist steps from the "Installation and Migration Checklist" on page 79.

Changing DB2 Installation Parameters

1 ***Evaluate and select appropriate values.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

The following values are provided as guidance information in establishing a DB2 system for use with OAM. If you have other applications sharing a DB2 system with OAM, you should evaluate the following and select values appropriate for all applications.

Note: These values are only suggestions and are given only for installation. You may need to change them for optimum performance.

Buffer Pools and Max Connects

When you defined DB2 job DSNTIJUZ, you specified values for the following storage sizes installation parameters. Evaluate the values you specified and ensure that your selected values will provide optimum performance. Change the values as required.

```
MAX USERS         (NUMCONCR)  200
MAX TSO           (NUMCONTS)  100
MAX BATCH         (NUMCONBT)  100
MIN BP0 BUFFERS   (BUFMIN00)  200
MAX BP0 BUFFERS   (BUFMAX00)  300
MIN BP1 BUFFERS   (BUFMIN01)  200
MAX BP1 BUFFERS   (BUFMAX01)  300
MIN BP2 BUFFERS   (BUFMIN02)  100
MAX BP2 BUFFERS   (BUFMAX02)  200
MIN BP32K BUFFERS (BUFMIN32)   50
MAX BP32K BUFFERS (BUFMAX32)  100
```

Note: The values may be verified or modified via DB2 dialog installation, or by DB2 Job DSNTIJUZ directly.

Refer to *DB2 for OS/390 Administration Guide* for additional information on storage sizes installation parameters.

EDM Pools

If the environmental descriptor management (EDM) function pool size is not large enough for the databases, DB2 errors are received. The following is suggested for EDM pool size:

```
EDM POOL      9 000    (20 storage groups)
EDM POOL     20 000   (100 storage groups)
```

Refer to *DB2 for OS/390 Administration Guide* for an explanation of EDM pool size.

IMS Resource Lock Manager (IRLM) Installation Parameters

As the size of objects used by your installation increases, it may be necessary to update DB2 locking parameters to avoid time-out conditions during the OSMC storage management cycle. The number of pages required by DB2 to store the object data increases with the increase in object size and this directly affects the number of page locks which will be required by DB2 when operations are performed for the object. If insufficient page locks are available, lock escalation occurs and will cause a time-out condition during the OSMC storage management cycle. You must tune the DB2 parameter which controls the number of page locks to prevent lock escalation.

The current DB2 default for the number of page locks per table space is 1000. This is the recommended minimum value for OAM usage; however, as object sizes increase you will need to increase this value beyond the default to prevent lock escalation. As object sizes approach 50MB, this value may need to be increased upwards near 1700 page locks per table space and possibly beyond. You may want to set this value initially to 1700 and perform further tuning as it is required for your particular installation.

Note: These lock calculations may need to be increased in a DB2 data sharing environment, especially during OSMC processing.

The DB2 parameter which must be updated is:

```
LOCKS PER TABLE(SPACE)===>
```

The value for this DB2 parameter can be specified in several ways:

- On IRLM Panel 2: DSNTIPJ, which is presented when you run the DB2 installation CLIST (DSNTINST).
- On Locking Update Panel: DSNTIPK, which can be selected from the Update DB2 Panel (DSNTIPB) when you select “U” (update) item 1 on the Main Panel (DSNTIPA1).
- Directly updating the DSNZPARM macro DSN6SPRM value for NUMLKTS in job DSNTIJUZ.

Refer to *DB2 for OS/390 Administration Guide* for additional information on DB2 locking parameters.

Date and Time Routines: OAM does not require that dates and times be in a particular format; however, OAM returns and displays dates and times only in ISO format. The following example shows the ISO date and time format returned by OAM:

```
YYYY-MM-DD
```

For more information regarding date and time formats, see *DB2 for OS/390 Administration Guide*.

DB2 Group Buffer Pools: Add the buffer pool information from “Buffer Pools and Max Connects” on page 83 or the buffer pools used for the OCDB, OAMADMIN database, object directories, and object storage databases to the CFRM policy. This only needs to be done if you are setting up an OAMplex and data sharing environment. For more information, see *DB2 for OS/390 Administration Guide* and *z/OS MVS Setting Up a Sysplex*.

Changing CICS Installation Parameters

If you plan to run OAM under CICS, make the following changes to your CICS installation before using OAM:

2 ***Update or create CICS PLT.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

- If you have a program list table (PLT) to specify programs to be executed in the post-initialization phase of CICS startup, add the following entry:

  ```
  DFHPLT TYPE=ENTRY,PROGRAM=CBRICONN
  ```

  This names the OSR CICS initialization load module (CBRICONN) and invokes CBRICONN when CICS is initialized.
- If you do not have a program list table (PLT) used to specify programs for execution in the post-initialization phase of CICS startup, use one of the following to generate your CICS PLT:

```
CICS Release 2 and prior              CICS Release 3

DFHPLT TYPE=INITIAL,SUFFIX=xx         DFHPLT TYPE=INITIAL,SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=CBRICONN    DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL                     DFHPLT TYPE=ENTRY,PROGRAM=CBRICONN
                                      DFHPLT TYPE=FINAL
END                                   END
```

For additional information on DFHPLT, refer to *CICS Resource Definition Guide*.

3 ***Update CICS PPT.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Add the following entry to the CICS program properties table (PPT):

```
CBRICONN DFHPPT TYPE=ENTRY,PGMLANG=ASSEMBLER,PROGRAM=CBRICONN,RES=YES
```

This statement adds the OSR CICS initialization load module (CBRICONN) to the PPT.

If you just created the PLT in the previous step, then add the following entry to the CICS PPT:

```
DFHPPT TYPE=ENTRY,PROGRAM=DFHPLTxx
```

For *xx*, substitute the suffix of the DFHPLT module having the entry for CBRICONN.

Note: Remember, if OAM was previously installed, the entry for the CBRHTRAN load module must be removed from the CICS PPT, and the entry for the OSMC transaction must be removed from the CICS PCT.

4 ***Update CICS SIT.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Add the following entry to the CICS system initialization table (SIT) to supply initialization parameters to CICS:

```
CICS Release 2 and prior

        EXITS=YES,      ****     USER EXIT INTERFACE         X
        PLTPI=vv,       *** POSTINITIALIZATION CONNECTION
        PPT=(yy,COLD)   **** SAMPLE PGMS + BASIC FACILITIES

CICS Release 3

        PLTPI=vv,       *** POSTINITIALIZATION CONNECTION
        PPT=(yy,COLD)   *** SAMPLE PGMS + BASIC FACILITIES
```

where:

vv Specifies the suffix of the DFHPLT module

yy Specifies the suffix of the DFHPPT module

5 ***Connect DB2 to CICS.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Connect DB2 to CICS using the procedure in the section "Connect DB2 to CICS (Optional)" in *DB2 for OS/390 Administration Guide*.

6 *Update DB2/CICS RCT.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

Add the following entry to the DB2/CICS resource control table (RCT).

Note: If the DSNCRCT entry for the plan CBRIDBS already exists as part of the installation of another product, such as ImagePlus, there is no need to repeat it in this installation of OAM. However, you must verify it to make certain the suggested thread values discussed in the warning below are consistent for use within an OAM installation. If the statement does not currently exist, follow the procedures to add the entry.

This statement specifies that CICS transaction *xxxx* accesses DB2 databases using plan CBRIDBS:

```
DSNCRCT TYPE=ENTRY,TXID=xxxx,THRDM=4,THRDA=4,THRDS=4,PLAN=CBRIDBS,   X
               ROLBE=NO,TWAIT=YES
```

where:

TXID=*xxxx*	Specifies the name you have given your transaction
THRDM=4	Suggested value, given only for installation
THRDA=4	Suggested value, given only for installation
THRDS=4	Suggested value, given only for installation

Attention: The values for THRDM, THRDA, and THRDS are only initial suggestions. These values must be adjusted with other related DFHSIT parameters for CICS, also with the related suggestions for DSNCRCT for DB2. After setting the initial values, you will need to adjust the number of active threads, maximum active transactions per class, and related parameters to obtain the best overall performance for your installation. If the described adjustments to the thread values in the DSNCRCT for DB2, transaction classes, and priorities in CICS are not made, CICS transactions using OSREQ may have very poor performance, or may completely block CICS-to-DB2 activities, causing CICS to wait and requiring CICS to be stopped and restarted for recovery. Refer to *DB2 for OS/390 Administration Guide* for more information.

Note: Remember that if OAM was previously installed, the OSMC transaction must be removed from the DB2/CICS RCT for MVS/DFP 3.3.1 (including MVS/DFP 3.3.1 OAM Object Tape PRPQ) and DFSMS 1.1.0. For information on the DB2/CICS RCT (DSNCRCT), refer to *DB2 for OS/390 Administration Guide*.

7 *Copy CBRICONN to DFHRPL.*

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

CBRICONN must be contained in a load library in the DFHRPL concatenation. CBRICONN is delivered in SYS1.LINKLIB. You may add SYS1.LINKLIB to your DFHRPL concatenation or copy CBRICONN into another load library in the concatenation. If you choose not to use SYS1.LINKLIB in your DFHRPL list, you must remember to upgrade the copy of CBRICONN every time the module in SYS1.LINKLIB is changed due to maintenance or a new release of OAM.

Modifying the Installation Exit to Handle Deleted Objects

8 ***Evaluate and implement the auto-delete installation exit.***

| *Perform this step* **only** *if you are running OSMC for expiration processing.*

One of the rules defined in the management class is the end of an object's life. OSMC can delete an object when its lifetime expires. An object can also expire through an explicit expiration date. If an object has an explicit expiration date, the explicit expiration date takes precedence over the management-class-defined lifetime. OSMC calls the auto-delete installation exit before it deletes any object. The auto-delete installation exit indicates by return code whether the object should be deleted. Also, the installation exit can record the deletion of an object so applications can be kept synchronized with the OAM object directory table. In an OAMplex, you should synchronize the instances of CBRHADUX across the OAMs to avoid one OAM deleting an object with the approval of the exit when there is another exit on another OAM that is set to deny the delete request.

| **Note:** The sample auto-delete installation exit now prevents objects from being deleted. If your previous installation of OAM relied on the sample auto-delete installation exit to allow objects to be automatically deleted, you now must modify the code in this exit to continue automatic deletion. You must also modify the code in this exit to define or change your installation handling of deleted objects. For more information about the installation exit, see "Auto-Delete Installation Exit (CBRHADUX)" on page 509.

Changing System Libraries

After using SMP/E to install DFSMSdfp, change the system libraries using the following procedures. Some procedures are completed only if the installation uses optical storage. These procedures are identified in the text. Some procedures are completed only if the installation will be using object tape support. These procedures are also identified in the text. Unless otherwise noted, all other procedures must be completed.

9 ***Update PARMLIB.***

| **9a** ***Update IGDSMSxx PARMLIB member.***

| *Perform the following steps if you want to automatically start the OAM address space during IPL. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices), and do* **not** *start the OAM address space for object support processing.*

| 1. Update PARMLIB member IGDSMS*xx* to include the OAM-related keywords:

| **OAMPROC***(procname)*
| Optional parameter that specifies the procedure name to be used to start the OAM address space. Specify this keyword to start the

OAM address space automatically during IPL. The procedure name is from 1 to 8 characters, and there is no default.

OAMTASK*(taskid)*
: Optional parameter that specifies the identifier to be used to start the OAM address space. If you specify this keyword without the OAMPROC keyword, it is ignored. This identifier is from 1 to 8 characters, and there is no default. Code the OAMTASK keyword if you prefer to use an identifier other than the *procname* when starting the OAM address space. The *taskid* is the identifier that is used on the START command. The *taskid* corresponds to the *identifier* parameter of the MVS START command documented in the *z/OS MVS System Commands*. See “Starting OAM” on page 221 for examples of the OAM START command.

DB2SSID*(ssid)*
: Optional parameter that specifies the name of the DB2 subsystem. OAM and ISMF use the specified DB2 subsystem name to connect to an appropriate DB2 subsystem. The subsystem name is from 1 to 4 characters, and there is no default.

The DB2SSID parameter is considered optional. If it is not specified, the system prompts the user for a specification when object storage groups are in the current configuration. Indicating “NONE” for this parameter is acceptable; however, doing so allows OAM to be initialized without the DB2 subsystem connection or access to the configuration database. This may be appropriate for a tape library-only environment, but if object support is required within the installation, a DB2 subsystem name other than “NONE” **must** be supplied.

Note: If a DB2 group name is specified instead of a DB2 subsystem name, which is supported by DB2 in a data sharing environment, the DB2 subsystem startup ECB option is not supported by DB2. In this situation, the DB2 subsystem must be started before OAM is initialized.

The optional parameters, OAMPROC and OAMTASK, are used only when the OAM address space is to be started automatically as part of SMS initialization. OAM can be used as the procedure name, the task identifier, or both. For more information concerning this PARMLIB member and all its associated keywords, refer to *z/OS DFSMSdfp Storage Administration Reference*.

9b ***Update IEFSSNxx PARMLIB member.***

Do **not** *perform this step at initial installation.*

1. Delete the *yyy*,CBRINIT entry in the IEFSSN*xx* member.

   where:

   yyy Is either OSM, or OAM.

You must verify or perform this step if you determine it has not yet been completed in your current environment.

2. Add the OAM1 entry in the IEFSSN*xx* member:

```
OAM1,CBRINIT,'[TIME=GMT][,MSG=xx][,OTIS=x][,UPD=x]'
```

where:

TIME=GMT Specifies that the time-stamp value in the object directory in DB2 will be based on GMT. If this option is omitted, or if any value other than GMT is specified, the object directory time stamp in DB2 will be based on local time.

MSG=*xx*
: Specifies the format for how the OAM message text appears:
- MSG=EM specifies that the message text is in mixed-case English. This is the default.
- MSG=EU specifies that the message text is in uppercase English.

OTIS=*x*
: Specifies whether OTIS should wait for JES to completely initialize before OTIS is started:
- OTIS=Y specifies that OTIS will not start until JES is completely initialized.
- OTIS=N specifies that OTIS will start independently from JES. This is the default.

UPD=*x*
: Specifies whether DB2 updates for the pending action date (ODPENDDT) and the last reference date (ODLREFDT) fields should be performed:
- UPD=Y specifies that the ODPENDDT and ODLREFDT fields should be updated on all OSREQ retrieves. This is the default.
- UPD=N specifies that the ODPENDDT and ODLREFDT fields should not be updated for any OSREQ retrieves.

Note: If UPD=N is used, it will not be possible to base transition criteria on the time since last use parameter in the ISMF management class definition.

OAM1 is the name of the subsystem, and CBRINIT is the name of the OSR initialization module executed at IPL time.

Note: The OAM1 entry must follow the SMS entry. The OAM1 entry defines the OAM1 subsystem; you must add this entry even if you do not plan to start the OAM address space. To prevent a possible system abend, make certain that the subsystem name in the step above is different from the PROCLIB member used to start OAM.

| **9c** ***Update SMFPRMxx PARMLIB member.***

You may perform this step at initial installation. For migration, you may verify or perform this step if you determine that it has not yet been completed in your current environment.

1. Verify and update if necessary the SMFPRM*xx* PARMLIB member to ensure it has been set up appropriately. For more information on this PARMLIB member, see “Changing SMF Recording” on page 506.

| **9d** ***Update IEAICSxx PARMLIB member.***

Perform this step at initial installation.

Defining Report Performance Groups

In order for RMF to collect statistics about a given OAM transaction name, a report performance group must be defined in the IEAICS*xx* member of PARMLIB for each transaction for which the installation wishes SRM and RMF to collect statistics.

To define a report performance group, a SUBSYS specification must be added to the IEAICS*xx* member of PARMLIB. Figure 13 on page 90 is an example of a SUBSYS specification that defines a report performance group for some of the

OAM transaction names associated with the OSREQ transaction class. For more information on the IEAICS*xx* member of PARMLIB, refer to *RMF User's Guide*.

```
SUBSYS=OAM,                          /* OAM REPORT PERF GROUP     */
  TRXCLASS=OSREQ,RPGN=400            /* OSREQ TRANSACTION CLASS   */
  TRXNAME=ACCESS,RPGN=401            /* OSREQ ACCESS              */
  TRXNAME=STOROPT,RPGN=421           /* OSREQ STORE TO OPTICAL    */
  TRXNAME=QUERY,RPGN=409             /* OSREQ QUERY               */
  TRXNAME=RETRVPO,RPGN=414           /* OSREQ RETRIEVE PRIMARY OPT*/
  TRXNAME=CHANGE,RPGN=402            /* OSREQ CHANGE              */
  TRXNAME=DELETEO,RPGN=405           /* OSREQ DELETE FROM OPTICAL */
  TRXNAME=UNACCESS,RPGN=423          /* OSREQ UNACCESS            */
```

Figure 13. IEAICSxx SUBSYS Specification for OSREQ Transactions

A complete OAM ICS SAMPLIB member is distributed with OAM. The name of the SAMPLIB is CBRICS00. This member defines a report performance group for each OAM transaction name. This SAMPLIB member can be copied into the IEAICS*xx* MEMBER of PARMLIB and edited as necessary.

9e ***Update CONSOLxx PARMLIB member.***

You may perform this step at initial installation. For migration, you may verify or perform this step if you determine that it has not yet been completed in your current environment.

The Action Message Retention Facility (AMRF) must be active in order for messages with descriptor code 3 (eventual action required) to be recalled through the use of the `D R,L,KEY=OAM` command after the messages have rolled off the MVS console. AMRF is activated at IPL by setting keyword `AMRF=Y` in the CONSOL*xx* PARMLIB member.

10 ***Create or Update CBROAMxx PARMLIB members.***

You must perform this step if you are using object tape support, setting up an OAMplex, customizing your OSMC support, or customizing your optical environment. Do **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices), or if you are running OSMC for expiration processing.*

This CBROAM*xx* member is used in conjunction with the OAM=*xx* parameter in the PROCLIB and must be created or updated to support optical or object tape storage. Default parameters are available for use within an optical environment. If you do not specify specific parameters, the defaults are used. The CBROAM*xx* PARMLIB member can now be included in any data set that is specified in the concatenation list in SYS1.IPLPARM(LOAD*xx*).

For information concerning this PARMLIB member and object tape support, see "SETOAM Statements for Object Tape Support" on page 91. For information concerning this PARMLIB member within an optical environment, see "SETOPT Statements for Use in an Optical Environment" on page 105. For information concerning this PARMLIB member in exploiting multiple object backup support, see "SETOSMC Statements for Use in the OSMC Environment" on page 109. For information concerning this PARMLIB member and Parallel Sysplex support, see "OAMXCF Statements in an OAMplex" on page 111.

SETOAM Statements for Object Tape Support

A CBROAM*xx* PARMLIB member contains SETOAM statements and is processed during OAM address space initialization to establish the tape related values for the object tape support. Creating or updating the CBROAM*xx* PARMLIB member with SETOAM statements is required for invocation of object tape support within your environment. If the SETOAM statement does not assign TAPEUNITNAME values to the Object Backup storage groups, then the backup copies of the objects are stored on optical media. If there are no SETOAM statements in the CBROAM*xx* member of PARMLIB, then the objects are stored on optical media by default.

Because an installation may want to tailor its object tape support for different initializations of OAM, multiple CBROAM*xx* members may be created. In addition, multiple SETOAM statements may be supplied in one CBROAM*xx* PARMLIB member. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for information on changing the SETOAM values dynamically, or on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

The SETOAM statement is used to determine whether backup objects are stored on tape volumes when they are written to an Object Backup storage group. If PARMLIB member, CBROAM*xx* may contain one or more SETOAM statements. If the same parameter is specified multiple times on the same SETOAM statement, the last occurrence of the parameter is accepted. If the same parameter is specified multiple times on different SETOAM statements, the last occurrence on the last statement is accepted. If any syntactical errors are encountered in processing the statements in the CBROAM*xx* member of PARMLIB, OAM issues a message, and the OAM address space will not successfully initialize.

Figure 14 on page 92 is an example of a CBROAM*xx* PARMLIB member that can be used as a sample for your installation. See the syntax diagrams that follow Figure 14 on page 92 for graphical depictions of the SETOAM statement. The descriptions of the keywords are found in the discussion of the SETOAM statement on "SETOAM Keyword Definitions" on page 93.

```
SETOAM DATACLASS(INMTL)
       DEMOUNTWAITIME(120)
       MAXTAPERETRIEVETASKS(2)
       MAXTAPESTORETASKS(2)
       MOUNTWAITIME(5)
       TAPECAPACITY(CST18 55555)
       TAPECAPACITY(CST36 88888)
       TAPECAPACITY(ECCST 99999999)
       TAPECAPACITY(ESOTERIC1 7654321)
       TAPECAPACITY(ESOTERIC2 77766655)
       TAPECAPACITY(ESOTERICn 2147483646)
       TAPEEXPIRATION(2035/165)
       TAPEFULLTHRESHOLD(4096)
       STORAGEGROUP(GROUP00
         DATACLASS(TAPEGRP)
         DEMOUNTWAITTIME(120)
         MAXTAPERETRIEVETASKS(2)
         MAXTAPESTORETASKS(1)
         TAPECOMPACTION
         TAPEDRIVESTARTUP(9999)
         TAPEEXPIRATION(2055/003)
         TAPEFULLTHRESHOLD(2048)
         TAPEPERCENTFULL(76)
         TAPEUNITNAME(3490))
```

Figure 14. CBROAMxx PARMLIB Member Sample

The syntax diagrams below show the syntax for the SETOAM statement. The first diagram shows the syntax for OAM global level parameters. The second diagram shows the subparameters for the STORAGEGROUP keyword parameter. For information regarding how to read syntax diagrams, see “How to Read Syntax Diagrams” on page xvii.

SETOAM Statement Syntax: OAM Global Level Parameters

```
►►──SETOAM──┬─────────────────────────────┬──┬────────────────────────────────┬──────────►
            └─DATACLASS(SMS data class)─┘  └─DEMOUNTWAITTIME(seconds)─┘

►──┬─────────────────────────────────┬──┬──────────────────────────────┬──────────────────►
   └─MAXTAPERETRIEVETASKS(tasks)─┘  └─MAXTAPESTORETASKS(tasks)─┘

►──┬─────────────────────────┬──┬─────────────────────────────────────┬────────────────────►
   └─MOUNTWAITTIME(minutes)─┘  └─TAPECAPACITY(unitname kilobytes)─┘

►──┬─────────────────────────────┬──┬──────────────────────────────────┬──────────────────►
   └─TAPEEXPIRATION(YYYY/DDD)─┘  └─TAPEFULLTHRESHOLD(kilobytes)─┘

►──┬───────────────────────────────────────────────────────┬──────────────────────────►◄
   └─STORAGEGROUP(See STORAGEGROUP Subparameters)─┘
```

SETOAM Statement Syntax: STORAGEGROUP Subparameters

```
►►──TAPEUNITNAME(unit-name)──┬─────────────────────────────┬──────────────────────────►
                             └─DATACLASS(SMS data class)─┘
```

```
►─┬──────────────────────────┬─┬──────────────────────────────┬──────────►
  └─DEMOUNTWAITTIME(seconds)─┘ └─MAXTAPERETRIEVETASKS(tasks)─┘

►─┬───────────────────────────┬─┬──────────────────┬──────────────────────►
  └─MAXTAPESTORETASKS(tasks)─┘ ├─TAPECOMPACTION───┤
                               └─NOTAPECOMPACTION─┘

►─┬───────────────────────────────────────────────┬─┬──────────────────────────┬─►
  └─TAPEDRIVESTARTUP(threshold in megabytes)─┘ └─TAPEEXPIRATION(YYYY/DDD)─┘

►─┬──────────────────────────────┬─┬──────────────────────────┬──────────►◄
  └─TAPEFULLTHRESHOLD(kilobytes)─┘ └─TAPEPERCENTFULL(percent)─┘
```

SETOAM Keyword Definitions

DATACLASS(*name*)

An optional parameter specifying the SMS data class to be used for Object and Object Backup storage groups that do not have their own DATACLASS specification on the STORAGEGROUP subparameter of the SETOAM statement. IF DATACLASS is specified at the OAM global level, but not at the storage group level, this specification of DATACLASS applies to each of the storage groups with a corresponding SETOAM statement which do not explicitly specify a DATACLASS. Using the DATACLASS keyword on the SETOAM statement at the global level allows the installation to affect Tape Device Selection Information (see the discussion on page 22 for details on TDSI), and volume expiration date for those Object or Object Backup storage groups which do not have an explicit DATACLASS keyword in their STORAGEGROUP subparameter list. There is NO global level OAM default DATACLASS.

It is not recommended that you allow ACS routines to assign or change the data class assignment of an OAM tape volume. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information at the storage group level or at the OAM global level, and it best suits the requirements for the tape volume that is being allocated. Allowing ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). It is the installation's responsibility to ensure that their ACS routines are written so as to not alter the data class construct for OAM tape volumes.

DEMOUNTWAITTIME(*seconds*)

An optional parameter that specifies the time, in seconds, that OAM waits before demounting and deallocating a tape drive that OAM is currently not using. For *seconds*, specify a decimal number between 1 and 9999. When the time interval expires, OAM rewinds and unloads the currently mounted tape cartridge and demounts and deallocates the device. The default for this optional parameter is 120 seconds.

There are some circumstances that may affect how this parameter is enforced:

- If a new tape drive allocation request comes in and OAM has already used the maximum number of tape drives (MAXTAPERETRIEVETASKS + MAXTAPESTORETASKS), then OAM ignores the DEMOUNTWAITTIME and deallocates this drive in order to allocate another.

- If OAM is canceled, a DEMOUNT occurs, and DEMOUNTWAITTIME is ignored.
- If a request to vary the drive offline is sent while the DEMOUNTWAITTIME is in effect for that drive in an OAM session, the drive cannot vary offline until the specified DEMOUNTWAITTIME elapses.
- If OAM finishes reading and writing to a tape before the DEMOUNTWAITTIME elapses, a demount, unload, rewind, or release of the allocated drive cannot take place until the specified DEMOUNTWAITTIME is complete.

MAXTAPERETRIEVETASKS(*tasks***)**

An optional parameter that specifies the maximum number of tasks within the OAM address space that can concurrently read objects from tape. This parameter controls the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading object data from tape. This includes the allocation of tape drives for processing the following requests:

- OSREQ RETRIEVE requests, where the primary copy of the object being retrieved is stored on tape
- OSREQ RETRIEVE requests, where VIEW=BACKUP and the backup copy is on tape
- Requests to read the primary copy of an object from tape during the OSMC storage management cycle
- Requests to read the backup copy or the primary copy of an object from tape during the execution of the OSMC optical volume recovery utility
- single object recovery
- OSREQ retrieve requests where the primary copy is on optical, but volume is not readable, access to backup is activated and the backup copy is on tape.

If there is one or more OAM application retrieving objects from multiple Object storage groups, and the primary copies of the objects are being retrieved from tape volumes, then the number or tasks specified with this parameter should be greater than or equal to the maximum number of Object storage groups being read from concurrently. This will eliminate the need to constantly mount and demount tapes belonging to different Object storage groups to satisfy OSREQ RETRIEVE requests.

The default is 1, which allows at least one task to be attached for processing read requests from tape volumes. This default allows groups previously writing objects to tape, but no longer having an explicit SETOAM statement in the CBROAM*xx* member, to be able to retrieve their previously written objects so long as a CBROAM*xx* PARMLIB member was successfully processed during OAM initialization. A CBROAM*xx* PARMLIB member must be successfully processed during OAM initialization for OAM object tape support to be active on the system.

MAXTAPESTORETASKS(*tasks***)**

An optional parameter that specifies the maximum number of tape drives used for writing objects to tape volumes. This includes the processing for the following:

- OSREQ STORE requests where the primary copy of the object is to be stored on tape
- OSMC class transition processing where the primary copy of the object is to be stored on tape
- Writing of backup copies of objects during the OSMC storage management cycle.
- Single object recovery

- Producing backup copies if backups are directed to tape
- MOVEVOL for backup copies if objects are directed to tape.

This parameter and the MAXTAPERETRIEVETASKS parameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space. For *tasks*, specify a decimal number between 1 and 100. The number specified for *tasks* with the MAXTAPESTORETASKS parameter should be greater than the number of Object storage groups that OAM applications are using for storing objects to tape volumes. If the number is less than the number of Object storage groups for the OAM applications, OAM will be frequently mounting and demounting tape volumes belonging to different Object storage groups to satisfy OSREQ STORE requests. Also, if objects are being written to the Object Backup storage group during the storage management cycle for multiple storage groups, and the backup copies are being written to tape volumes, then the number of tasks specified with this parameter should encompass this activity. Note that the maximum number of concurrent storage groups, which may or may not all be causing objects to be written to the Object Backup storage group, is specified by MAXS= in the PARM field on the JCL EXEC statement in the OAM cataloged procedure. If this parameter is not specified on any SETOAM statement, OAM sets the default to 1.

Note: You should verify that there are enough tape drives (specified in the TAPEUNITNAME keyword) available to handle the values assigned to the MAXTAPESTORETASKS and MAXTAPERETRIEVETASKS parameters of the SETOAM statement. How the drives are spread among the Object storage groups or the Object Backup storage group depends on the values assigned to the MAXTAPESTORETASKS, MAXTAPERETRIEVETASKS, TAPEDRIVESTARTUP (see the discussion on page 102 for details on TAPEDRIVESTARTUP), and the TAPEUNITNAME parameters for each Object and the Object Backup storage group. If there are not enough tape drives, and the recovery logic for allocation is not able to obtain a tape drive for this request, then the request fails.

MOUNTWAITTIME(*minutes***)**

An optional parameter that specifies the time, in minutes, that OAM waits for a tape volume to be mounted. For *minutes*, specify a decimal number between 1 and 120. When this interval expires, OAM issues message CBR6405D to the operator asking if the tape volume can be mounted. If the response is “Y”, OAM resets the timer for another *minutes* interval. If the next interval expires and the tape volume is still not mounted, the same message is sent to the operator. If the reply is “C” and the mount was for tape that was to be read, OAM ends the task, fails the request to retrieve the object, and the volume is marked as “lost”. If the operator replies “C” to this message and the mount was for a tape volume that was the target of a nonspecific (grouped) write request, a message is issued, and the volume is marked “lost” in the OAM internal control blocks, and the write request is retried on another volume. This processing applies to stand-alone, automated tape library dataservers and manual tape libraries. If you do not specify this parameter on any SETOAM statement, the OAM default is five minutes. If this is a mount for an MVS scratch tape, and retry is attempted, the request fails.

Note: You may issue the F OAM,DISPLAY,LOSTVOL command to determine the volume serial number of each lost volume, both tape and optical. See “Displaying Volumes that Have LOSTFLAG Set” on page 301 for further information on this command.

TAPECAPACITY(*unitname kilobytes***)**

An optional parameter specifying a unit name with a numeric value in kilobytes from 1 to 2 147 483 646. This parameter allows a user to specify a tape capacity for tapes written using a general recording technology, as well as a different capacity for tapes written to drives associated with esoteric unit names. This parameter indicates the tape capacity desired for the three general specifications (CST18, CST36, or ECCST) and esoteric unit name specifications. All esoteric unit name specifications are verified as valid esoteric unit names that are defined to the system. The tape capacity specified on this parameter becomes the value at which OAM considers a tape volume filled.

If this parameter is not specified in the SETOAM statement, the following defaults will be used:

- Standard 18-track recording technology defaults for CST18
- Standard 36-track recording technology defaults for CST36
- Enhanced capacity recording technology defaults for ECCST

If the TAPECAPACITY parameter is specified as part of the SETOAM statement, but there is no specification for CST18, CST36, or ECCST, or there is no valid generic or esoteric unit names entered, OAM will not start and message CBR0325I is issued.

If a 3480, 3480x, or 3490 unit name ′is specifically used on this parameter, OAM accepts it, but considers it to be an esoteric unit name. Generally, the needs of these unit types are covered by the CST18, CST36, or ECCST keywords so they do not normally need to be specified; however, if they are specified, OAM accepts them and ensures that they are valid unit names. If the esoteric unit name used with this parameter matches the *tapeunitname* subparameter on the SETOAM STORAGEGROUP parameter, the tape capacity associated with the tapeunitname subparameter is used instead of any equivalent CST18, CST36, or ECCST specification for this parameter.

This parameter is used at the OAM global level; however, it is possible to specify a desired tape capacity at the storage group level. This can be done by specifying an esoteric unit name on the *tapeunitname* subparameter on the SETOAM STORAGEGROUP parameter along with the use of the *tapecapacity* parameter. For example, if esoteric TAPE1 is specified on the *tapecapacity* parameter (such as, TAPECAPACITY(TAPE1 5000000)), then when the SETOAM STORAGEGROUP(*storagegroupname* TAPEUNITNAME(TAPE1) is used on the same SETOAM statement, the storage group uses the tape capacity of 5 000 000.

If the tape capacity value in the Tape Volume Table is different than that specified on the *tapecapacity* parameter of the SETOAM statement, the value of the SETOAM statement is used for the duration of the OAM session, or until changed or deleted on the SETOAM statement. The Tape Volume Table capacity is not changed after it is initially set during the first time the volume was written to. This is to avoid a changing capacity and a volume fluctuating between full or not full based on a differing capacity. If it is determined that the capacity in the Tape Volume Table must be changed, SPUFI can be used to dynamically perform this update and OAM will accept it. The freespace and percent full calculations for the volume will be based on the updated capacity and will be recalculated when the volume is written to again as a partial volume, or when OAM is restarted. Although this value can be changed using SPUFI, this should be used with caution and at the risk of the user, because of the potential problems that might arise from updating the DB2 tables in this manner.

In releases prior to DFSMS/MVS 1.5.0, the capacity value was specified as millimeters. This value is now specified in kilobytes, so when OAM is first started after installing the DFSMS/MVS 1.5.0 support, all volumes with the millimeter values will be changed to reflect kilobytes for the capacity format. The following are the old and new defaults for tape capacities:

Type	Previous Support	DFSMS 1.5.0 Support
Standard 18-track	150 000 millimeters	218554 KB
Standard 36-track	300 000 millimeters	437109 KB
Enhanced Capacity	600 000 millimeters	874218 KB

After OAM comes up the first time with this new support, none of the volumes marked full will become non-full volumes and none of the non-full volumes will become full based on the value of the *tapecapacity* parameter of the SETOAM statement. Only the unit of measure (millimeter versus kilobytes) for the volume freespace changes; the actual amount of freespace on the volume remains the same. The percent full for the volume remains the same as previous releases. If previous existing tape capacities used prior to DFSMS 1.5.0 (150 000, 300 000, or 600 000 millimeters) were changed using SPUFI or any other tool, they will not be updated to the new kilobyte defaults. It is the responsibility of the installation to update these values as needed.

TAPEEXPIRATION(*YYYY/DDD***)**

An optional parameter specifying the year and date (*YYYY/DDD*) assigned to the data sets on OAM object tape volumes used for expiration purposes where:

- *YYYY* is a four-digit number that specifies a year from 1900 through 2155, and
- *DDD* is a three-digit number that specifies a day from 001 through 366.

The TAPEEXPIRATION date overrides the expiration date defined in the DATACLASS parameter for the data sets residing on the tape volume.

If you specify the TAPEEXPIRATION date for the data sets residing on the OAM object tape volume as the current date or a date preceding the current system date, the data sets are considered previously expired and are therefore eligible for immediate replacement. OAM issues a message to this effect (CBR0317I) to allow the user to change the TAPEEXPIRATION value in the SETOAM statement in the CBROAM*xx* PARMLIB member being used if necessary.

Expiration dates of 1999/365 and 1999/366 are considered “never-scratch” dates. Data sets with these expiration dates are not deleted or written over.

TAPEFULLTHRESHOLD(*kilobytes***)**

An optional parameter specifying a numeric value of 0 through 999 999 representing the number of kilobytes of available freespace allowed for any volume belonging to any object tape storage group in the configuration. When the number of kilobytes of freespace for a tape volume falls below the TAPEFULLTHRESHOLD parameter specified at the OAM global level, the volume is marked full and is not used for any further write requests. The default value for this parameter is zero.

It is important to select a threshold value the allows tape volumes to be marked full in a consistent manner. Consider the size of the objects stored, and if the size of the objects is consistent, select a threshold value that is slightly larger than that size. If volumes are not being selected for new objects and they are not being marked full, increase the value of this parameter.

During OAM initialization, the tape volume full status is checked in conjunction with the TAPEFULLTHRESHOLD parameter (if specified at the global level) to determine the volume's freespace and the TAPEPERCENTFULL subparameter (if specified at the storage group level) to determine the volume's percent full status. The volume full status is changed from full to not full if:

- Freespace for the volume is greater than the TAPEFULLTHRESHOLD parameter value and the volume percent full value is less than the TAPEPERCENTFULL subparameter.

The volume full status is changed from not full to full if:

- Freespace for the volume is less than or equal to the TAPEFULLTHRESHOLD parameter value or the volume percent full value is equal to or greater than the TAPEPERCENTFULL subparameter value.

STORAGEGROUP(*name***)**

An optional parameter that specifies tape related parameters for a specific Object or Object Backup storage group which is in the active configuration, and which was previously defined using the ISMF storage group application. This parameter on the SETOAM statement provides additional information beyond what was specified using ISMF for the Object or Object Backup storage group to which it pertains. At times, the information overrides what was specified using ISMF. For example, if the Object Backup storage groups has a valid SETOAM statement, then the backup copies of objects are written on tape regardless of the optical disk libraries supplied in the ISMF definition of the Object Backup storage group. For *name*, specify the name of an Object or Object Backup storage group. The parameter listed below applies to the name of the Object or Object Backup storage group that you specify.

TAPEUNITNAME(*unit-name***)**

A required subparameter of the STORAGEGROUP parameter that specifies the type of tape drive that OAM uses when writing data to or reading data from an Object or Object Backup storage group. This TAPEUNITNAME is the MVS unit name used by OAM to initially allocate a scratch tape when an object is stored to this Object or Object Backup storage group and stored on a tape volume. For *unit-name*, specify the name of a valid MVS esoteric (group of devices defined to a group name) or a generic unit name. Valid generic unit names are:

- 3480—a base 3480 device
- 3480X—a 3480 device with the IDRC feature, or a base 3490 device
- 3490—a 3490E device
- 3590-1—a 3590-1 device

Note: When OAM requests a mount for a generic tape with a TAPEUNITNAME of 3480, it accepts the 3480 tape drive chosen regardless of whether that tape drive has IDRC. MVS Allocation does not use the JCL/Dynamic Allocation parameter for compaction when determining device eligibility. If tape compaction is requested, and the tape is mounted on a 3480 tape drive that does not have IDRC, allocation fails. To prevent this failure, OAM does not allow tape compaction with a generic TAPEUNITNAME of 3480. OAM uses the NOTAPECOMPACTION keyword with all data for this TAPEUNITNAME.

The unit name specified is associated with each tape volume used for output during the process of writing objects to tape that belong to a specified Object or Object Backup storage group. This unit name is saved in the corresponding rows in the TAPEVOL table for each of these tape volumes, and is used during later allocations of these tape volumes for either reading or writing processing.

TAPEUNITNAME is a required keyword, and is specified for all allocations. In the automated tape library dataservers and manual tape libraries, this information may be used by the ACS filter routines, but is not required for device allocation. In the stand-alone environment, this information is **critical** in the allocation decision making process.

Note: Even though there is a tape unit name specified for the group, the ACS routines (for environment ALLOC), can override the TAPEUNITNAME specification by assigning the allocation to a Tape storage group, thereby steering the allocation into an ATLDS, or an MTL.

Grouping Acceptable Esoteric Unit Names: If an esoteric unit name is specified, the esoteric unit name must consist of only IBM 3480, 3490, 3490E, 3590-1, 3590-1 in 3490E emulation mode, 3590-E in 3590-1 emulation mode, or 3590-E in 3490E emulation mode. If an esoteric unit name is specified and is a mixed esoteric group consisting of at least one IBM 3480 tape drive without the IDRC feature **and** one IBM 3480 or base 3490 tape drive with the IDRC feature, then the TAPECOMPACTION keyword is ignored and the NOTAPECOMPACTION keyword is assumed. This guarantees that a tape volume written on any drive in the esoteric group can be read on any other drive in the same esoteric group.

Rejecting Incompatible Esoteric Unit Name Groups: To ensure that objects written on any drive in the esoteric group can be read on any drive in the same esoteric group, OAM does not allow a mixed esoteric unit name consisting of:
- 3590-E tape drives in 3590-1 emulation mode and 3590-E tape drives in 3490E emulation mode.
- 3590-1 tape drives and 3590-1 tape drives in 3490E emulation mode.
- 3590-1 tape drives and any other type of 3490E, base 3490, or 3480 tape drives.
- 3490E tape drives and any other type of base 3490 or 3480 tape drives.

If an esoteric unit name is specified and the esoteric unit name is a mixed esoteric group, as mentioned above, the SETOAM command will be rejected with an error message indicating that an esoteric unit name consisting of mixed devices is invalid.

Additionally, the installation should avoid using an esoteric unit name consisting of tape drives that write in different recording technologies such as:
- 3590-E tape drives in 3590-1 emulation mode (256-track recording technology) and 3590-1 tape drives (128-track recording technology), or
- 3590-E tape drives in 3490E emulation mode (256-track recording technology) and 3490E tape drives (36-track recording technology).

If the esoteric unit name mentioned above is associated with the Object or Object Backup storage group for which the read/write is being performed, the read/write may fail if an incompatible tape drive is selected by MVS allocation.

DATACLASS(*name*)
An optional subparameter of the STORAGEGROUP parameter that specifies the SMS data class to be associated with this Object or Object Backup storage

group. Usage of the DATACLASS keyword on the SETOAM statement allows an installation to affect things such as, TDSI, and the tape volume expiration date on an individual Object or Object Backup storage group level. If you do not specify DATACLASS on the SETOAM statement for a *specific* storage group, but you did specify DATACLASS at the global level of the SETOAM statement, then the global OAM DATACLASS specification applies to the specific storage group. If there is no DATACLASS specification at either level, then there is NO default for DATACLASS value used.

It is not recommended that you allow ACS routines to assign or change the data class assignment of an OAM tape volume. The data class for OAM tape volumes is determined by the SETOAM statement of the CBROAM*xx* PARMLIB member at MVS scratch tape allocation. The SETOAM statement provides this information at the storage group level or at the OAM global level, and it best suits the requirements for the tape volume being allocated. Allowing ACS routines to alter this specification could create unexpected consequences (for example, no compaction of the data when the SETOAM statement specified compaction). It is the installation's responsibility to ensure their ACS routines are written to not alter the data class construct for OAM tape volumes.

DEMOUNTWAITTIME(*seconds***)**

An optional subparameter of the STORAGEGROUP parameter. It specifies the time, in seconds, that OAM waits before demounting and deallocating a tape drive (allocated for the storage group specified with the STORAGEGROUP parameter), that OAM is currently not using. For *seconds*, specify a decimal number between 1 and 9999. When the time interval expires, OAM rewinds and unloads the currently mounted tape cartridge and demounts and deallocates the device. The default for this optional parameter is 120 seconds.

There are some circumstances that may affect how this parameter is enforced:

- If a new tape drive allocation request arrives and OAM has already used the maximum number of tape drives (MAXTAPERETRIEVETASKS + MAXTAPESTORETASKS), then OAM ignores the DEMOUNTWAITTIME and deallocates this drive in order to allocate another.
- If OAM is canceled, a DEMOUNT occurs, and DEMOUNTWAITTIME is ignored.
- If a request to vary the drive offline is sent while the DEMOUNTWAITTIME is in effect for that drive in an OAM session, the drive cannot vary offline until the specified DEMOUNTWAITTIME elapses.
- If OAM finishes reading and writing to a tape before the DEMOUNTWAITTIME elapses, a demount, unload, rewind, or release of the allocated drive cannot take place until the specified DEMOUNTWAITTIME is complete.

MAXTAPERETRIEVETASKS(*tasks***)**

An optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for reading objects from tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently read objects from tape for the storage group specified with the STORAGEGROUP parameter. This subparameter and the MAXTAPESTORETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for reading from and writing to tape volumes belonging to the specified Object or Object Backup storage group. For *tasks*, specify a decimal number between 1 and 100.

The value specified with the MAXTAPERETRIEVETASKS *subparameter* of the STORAGEGROUP parameter for a specific Object or Object Backup storage group cannot exceed the global maximum number of tape retrieve tasks specified with the MAXTAPERETRIEVETASKS *parameter* of the SETOAM statement. If it does, an error message is issued, and the SETOAM statement is rejected. If you do not specify this subparameter on any SETOAM statement, the OAM default is 1.

MAXTAPESTORETASKS(*tasks*)
An optional subparameter of the STORAGEGROUP parameter that specifies the maximum number of tape drives used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. This parameter specifies the maximum number of tasks within the OAM address space that can concurrently write objects to tape volumes belonging to the Object or Object Backup storage group specified with the STORAGEGROUP parameter. This subparameter and the MAXTAPERETRIEVETASKS subparameter control the maximum number of tape drives that can be concurrently allocated to the OAM address space for writing to and reading from tape volumes belonging to the specified Object or Object Backup storage group. For *tasks*, specify a decimal number between 1 and 100.

The value specified with the MAXTAPESTORETASKS *subparameter* of the STORAGEGROUP parameter for a specific Object or Object Backup storage group, cannot exceed the global maximum number of tape store tasks specified with the MAXTAPESTORETASKS *parameter* of the SETOAM statement. If it does, an error message is issued, and the SETOAM statement is rejected.

| **Note:** To use more than one tape drive for a storage group to write object data to tape, the TAPEDRIVESTARTUP threshold must be low enough to trigger the startup of the additional tape drive. This threshold is a value (in megabytes) of write data pending for this storage group. See the discussion concerning TAPEDRIVESTARTUP on page 102 for more information.

TAPECOMPACTION | NOTAPECOMPACTION
Mutually exclusive optional subparameters of the STORAGEGROUP parameter that specify the use of the compaction feature. These parameters specify whether the objects for this storage group are to be written in compacted or noncompacted format. See Table 20 on page 102 for an example of this selection process.

TAPECOMPACTION specifies that the compaction feature of the tape drive is enabled when OAM is writing objects to tape which belong to the specified Object or Object Backup storage group. This parameter is ignored if the unit name specified with the TAPEUNITNAME parameter is a mixed esoteric group. See “Rejecting Incompatible Esoteric Unit Name Groups” on page 99 for more detail concerning mixed esoteric groups.

NOTAPECOMPACTION specifies that the compaction feature of the tape drive is disabled when OAM is writing objects to tape belonging to the specified Object or Object Backup storage group.

If you do not specify this subparameter on any SETOAM statement, then the OAM default for the specified Object or Object Backup storage group is determined from the DATACLASS associated with this storage group. To utilize either the tape compaction or the no tape compaction feature, specify the DATACLASS with either TAPECOMPACTION or NOTAPECOMPACTION. If there is no DATACLASS associated with this storage group or if the DATACLASS associated with this storage group has a blank

TAPECOMPACTION specification, then whether tapes added to this storage group will use the compaction feature is determined by the DEVSUP parameter defaults provided during the allocation process.

Table 20. Example of the TAPECOMPACTION / NOTAPECOMPACTION Selection Process

Was TAPECOMPACTION or NOTAPECOMPACTION specified?	
YES	Do what is specified.
NO	Check data class specification.
Was DATACLASS specified?	
YES	Was TDSI compaction either YES or NO? **YES** Do what is specified. **NO** Do what is specified.
NO	DATACLASS was not specified, did not apply, or had a blank compaction specification. Consider the DEVSUP specification.
Is there a DEVSUP specification?	
YES	Do what is specified.
NO	If 3480, use NOTAPECOMPACTION. If 3480x or 3490, use TAPECOMPACTION. If 3590-1, use TAPECOMPACTION.

Note: If you want to change the TAPECOMPACTION | NOTAPECOMPACTION attribute associated with the storage group and you want OAM to only write data in the new format, you must update the tape volume table using the MODIFY OAM,UPDATE,VOLUME,*volser*,WRITABLE,N command (or by using SPUFI while the OAM address space is not active) to mark the existing tape volumes in the storage group unwritable. You can update the SETOAM TAPECOMPACTION parameter by using the MODIFY OAM,UPDATE,SETOAM,*scope*,TCOMP,Y command or by updating the CBROAM*xx* member of PARMLIB and restarting the OAM address space.

TAPEDRIVESTARTUP(*threshold in megabytes***)**

An optional subparameter of the STORAGEGROUP parameter that specifies the drive startup threshold used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. The parameter is used to indicate when OAM is to start the use of another tape drive for writing objects to tape volumes belonging to the storage group specified with the STORAGEGROUP parameter. When the number of MB of object data waiting to be written to tape is divided by the number of tape drives currently writing object data to tape exceeds the threshold specified by *threshold in megabytes*, OAM attempts to use another tape drive to write object data to the specified Object or Object Backup storage group.

However, the maximum number of tape drives being used by OAM to write object data to a specific Object or Object Backup storage group is limited by the value specified with the MAXTAPESTORETASKS subparameter of the STORAGEGROUP parameter on the SETOAM statement. The limit specified with this subparameter will never be exceeded.

Additionally, the maximum number of tape drives being used by OAM to write object data to all Object or Object Backup storage groups is limited by the value specified with the global MAXTAPESTORETASKS parameter of the SETOAM statement. The limit specified with this parameter will never be exceeded.

For *threshold in megabytes(MB)*, specify a decimal number between 1 and 9999. If you do not specify this subparameter on any SETOAM statement, the OAM default is 9999.

Note: Drive startup threshold in an optical environment is determined differently than the threshold in a tape storage environment. See the discussion concerning “DRIVE STARTUP THRESHOLD” on page 129 for more information.

TAPEEXPIRATION(*YYYY/DDD*)

An optional subparameter of the STORAGEGROUP parameter. It specifies the year and date (*YYYY/DDD*) assigned to the data sets on OAM object tape volumes belonging to a specific Object or Object Backup storage group for expiration purposes where:

- *YYYY* is a four-digit number that specifies a year from 1900 through 2155
- *DDD* is a three-digit number that specifies a day from 001 through 366

The TAPEEXPIRATION date for the data sets on the tape volumes belonging to the Object or Object Backup storage group overrides the expiration date defined in the DATACLASS parameter for these data sets.

If you specify the TAPEEXPIRATION date for the data sets on the tape volumes belonging to the specific Object or Object Backup storage group as the current date or a date preceding the current system date, the data sets are considered previously expired and are therefore eligible for immediate replacement. OAM issues a message (CBR0317I) to allow the user to change the TAPEEXPIRATION value in the SETOAM statement of the CBROAM*xx* PARMLIB member being used if necessary.

Expiration dates of 1999/365 and 1999/366 are considered “never-scratch” dates. Data sets with these expiration dates are not deleted or written over.

TAPEFULLTHRESHOLD(*kilobytes*)

An optional subparameter of the STORAGEGROUP parameter. It specifies a numeric value of 0 through 999999 representing the number of kilobytes of available freespace allowed for each volume belonging to the object tape storage group specified in the STORAGEGROUP parameter. When the number of kilobytes of freespace for a tape volume falls below the TAPEFULLTHRESHOLD subparameter for the storage group to which that volume belongs, the volume is marked full and is not used for any further write requests. The default value for this parameter is zero.

It is important to select a threshold value that allows tape volumes to be marked full in a consistent manner. Consider the size of the objects stored, and if the size of the objects is consistent, select a threshold value for the storage group that is slightly larger than that size. If volumes are not being selected for new objects and they are not being marked full, increase the value of this parameter.

During OAM initialization, the tape volume full status is checked in conjunction with the TAPEFULLTHRESHOLD subparameter to determine the volume’s freespace and the TAPEPERCENTFULL subparameter to determine the volume’s percent full status. The volume full status is changed from full to not full if:

- Freespace for the volume is greater than the TAPEFULLTHRESHOLD subparameter value and the volume percent full value is less than the TAPEPERCENTFULL subparameter.

The volume full status is changed from not full to full if:

- Freespace for the volume is less than or equal to the TAPEFULLTHRESHOLD subparameter value or the volume percent full value is equal to or greater than the TAPEPERCENTFULL subparameter value.

TAPEPERCENTFULL(*percent*)

An optional subparameter of the STORAGEGROUP parameter that specifies the percent full utilization used for writing objects to tape volumes belonging to a specific Object or Object Backup storage group. This parameter indicates at what percent of utilization OAM stops writing objects to tape volumes belonging to the storage group specified with the STORAGEGROUP parameter on the SETOAM statement.

When the tape volume utilization percentage for a tape volume belonging to an Object or Object Backup storage group reaches or exceeds the threshold specified by *percent*, OAM stops writing objects to the tape volume. The tape volume is marked full and another tape volume belonging to the specified Object or Object Backup storage group is selected for the continuation of writing objects to that storage group. If there are no tape volumes in the storage group with enough space to accommodate the object to be written, or if TAPEDRIVESTARTUP processing is attempting to start an additional tape drive and an additional scratch tape is required to start that drive, an OAM scratch tape is added to the group. If there are no OAM scratch tapes available, then an MVS scratch tape is requested and added to the Object or Object Backup storage group.

For *percent*, specify a decimal number between 1 and 100. If you do not specify this subparameter on any SETOAM statement, the OAM default is 100.

The TAPEPERCENTFULL subparameter is retroactive for the Object and the Object Backup storage groups defined in the ACDS. Whenever OAM is started and the TAPEPERCENTFULL for an Object or Object Backup storage group which is currently defined in the ACDS has been changed since the last time OAM was started, that new TAPEPERCENTFULL value is applied to all tapes which currently belong to the subject STORAGEGROUP. This means that several changes might be made to the tape volume table rows for the volumes in the storage group:

- The freespace (FRESPACE) for a volume might increase or decrease depending on whether the TAPEPERCENTFULL is increased or decreased.
- The volume full indicator (FULL) might change from full to not full, or from not full to full, depending on whether the TAPEPERCENTFULL is increased or decreased.

The adjustment is made to all affected tapes regardless of whether the tapes were previously marked full, unreadable, or unwritable. If new volumes are added to the storage group, they conform to the new TAPEPERCENTFULL value specified on the SETOAM statement that is being used for the current OAM initialization.

| **Note:** If you modify the TAPEPERCENTFULL value using the MODIFY
| OAM,UPDATE command, a volume you have marked full may be
| subsequently marked not full. This is because the volume's current tape
| full percentage is less than the value of the TAPEPERCENTFULL

parameter on the SETOAM statement. If you intend to mark the volumes in an Object or Object Backup storage group as full, then you must increase the value of the volume's percent full value (PFULL) to 100.

NOT Programming Interface information

Note: OAM can mark a tape volume full when:
- An 18-track tape reaches sector 69
- A 36-track tape reaches sector 1 on the second wrap. This is done to prevent OAM from falling into EOV processing.

End of NOT Programming Interface information

For optical storage, the volume full threshold parameter is used to determine the threshold value for an optical volume. See "VOLUME FULL THRESHOLD" on page 131 for information on the volume full threshold parameter for optical volumes.

SETOPT Statements for Use in an Optical Environment

The SETOPT statement and its associated keywords of the CBROAM*xx* PARMLIB member define rules at a global and storage group level used by OAM to administer optical support. Creating or updating the CBROAM*xx* PARMLIB member with SETOPT statements is required to support the returning of rewritable optical volumes to a common scratch pool or to reinitialize the volumes to their original storage group affiliation. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for information on changing the SETOPT values dynamically, or on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

Sample SETOPT Statement

Figure 15 shows examples of SETOPT statements in the CBROAM*xx* PARMLIB member that can be used as samples for your installation. The descriptions of each of the keywords are found in the discussion of the SETOPT statement (see "SETOPT Keyword Definitions" on page 106).

```
SETOPT OPTICALREINITMODE(GROUP)
       OPTICALDISPATCHERDELAY(4)
       MOUNTWAITTIME(3)
       STORAGEGROUP(GROUP01)
       STORAGEGROUP(GROUP01 OPTICALREINITMODE(GROUP))

              OR

SETOPT OPTICALREINITMODE(OAMSCRATCH)
       OPTICALDISPATCHERDELAY(3)
       SCRENTRYTHRESHOLD(5000)
       STORAGEGROUP(GROUP02)
       STORAGEGROUP(GROUP02 OPTICALREINITMODE(OAMSCRATCH))
```

Figure 15. CBROAMxx PARMLIB Member Samples using the SETOPT Statement and Optional Parameters

Note: If STORAGEGROUP is specified, then *object_storage_group* **must** be specified. All other parameters under STORAGEGROUP are optional.

The syntax for the SETOPT statement follows.

SETOPT Statement Syntax: OAM Global Level Parameters

```
►►──SETOPT──┬────────────────────────┬──┬─────────────────────────────────────┬──────►
            └─MOUNTWAITTIME(minutes)─┘  └─OPTICALDISPATCHERDELAY(seconds)─┘

►──┬───────────────────────────────────────────────┬──┬────────────────────────────┬──►
   └─OPTICALREINITMODE(reinitialization mode)─┘  └─SCRENTRYTHRESHOLD(value)─┘

►──┬────────────────────────────────────────────┬──────────────────────────────────►◄
   │                                       (1)  │
   └─STORAGEGROUP(object_storage_group──────)─┘
```

Notes:

1 See STORAGEGROUP Subparameters

SETOPT Statement Syntax: STORAGEGROUP Subparameters

```
                                                 (1)
►►──OPTICALREINITMODE(reinitialization_mode──────)──────────────────────────────►◄
```

Notes:

1 *reinitialization_mode* = GROUP or OAMSCRATCH

SETOPT Keyword Definitions

The following keywords are defined as they pertain to the OAM SETOPT statement.

MOUNTWAITTIME

A specified amount of time (in minutes) that can pass while a volume is waiting to be mounted on an operator-accessible drive within an optical library. After this time has expired, message CBR4426D is issued to allow the operator to retry or to cancel the volume mount request.

This value can be any numeric value from 1 to 9999. If the operator retries the mount request, the value specified in the MOUNTWAITTIME parameter is used for the retry. The default value of this parameter is five minutes.

Because this parameter can be changed dynamically, the new value assigned to the MOUNTWAITTIME is used for any currently executing volume mount request. The time of the initial mount request will be compared to the value specified in the MOUNTWAITTIME parameter and is deducted from the specified parameter value. For example, if a volume mount request has been waiting for one minute and the MOUNTWAITTIME is changed to indicate a wait time of two minutes from a previous value of three, the volume mount request will only have one more minute to finish before CBR4426D is issued. If a retry is requested after CBR4426D is issued, and then the MOUNTWAITTIME is dynamically changed after the reply to retry the request, the minutes specified in the MOUNTWAITTIME parameter are compared to the time of the most recent reply to this message to determine the exact wait time.

OPTICALDISPATCHERDELAY

Specifies the number of seconds that the OAM optical dispatcher is to delay processing of certain requests in order to minimize flipping of optical disk cartridges in an automated optical storage library. The OAM optical dispatcher will delay processing of a unit of work for a specific period of time, when ALL of the following conditions are true:

- A read request for an object on a currently mounted optical disk volume has just been completed.
- There is no request for the currently mounted optical disk volume waiting to be processed on the OAM optical dispatcher queue.
- The OAM optical dispatcher has found a read request for another optical disk volume (either the opposite side of the currently mounted volume or for an unmounted optical disk volume) and is about to dispatch this unit of work.
- A nonzero optical dispatcher delay value has been specified with the OPTICALDISPATCHERDELAY keyword on the SETOPT statement in the CBROAM*xx* PARMLIB member.

In this situation, the OAM optical dispatcher delays the dispatching of this selected unit of work (for the number of seconds specified by the installation) expecting that another read request for the currently mounted optical disk volume will arrive within this delay interval. The OAM optical dispatcher will delay dispatching of the selected unit of work for up to the number of seconds specified with the OPTICALDISPATCHERDELAY keyword on the SETOPT statement in the CBROAM*xx* PARMLIB member.

If another read request for the currently mounted optical disk volume arrives within the delay interval, that unit of work will be dispatched immediately upon arrival. If no read request for the currently mounted optical disk volume arrives within the delay interval another request for a different optical disk volume (either the opposite side of the currently mounted optical disk volume or an unmounted optical disk volume) is dispatched.

The value specified with the OPTICALDISPATCHERDELAY value can be used to circumvent a performance problem when IBM optical disk libraries (IBM 9246 and 3995 optical libraries) are used in conjunction with certain microfiche replacement applications. The problem involves the constant servicing of requests for data on both sides of an optical disk cartridge resulting in the cartridge being constantly flipped over to access data on the opposite side of the optical disk cartridge. This constant flipping of the cartridge results in longer response times for requests to read data from each side of the optical disk cartridge.

Valid value *seconds* specifies the number of seconds that the OAM optical dispatcher is to delay dispatching of specific units of work under the circumstances described above. Valid values for seconds is a decimal number between 1 and 60. If use of this parameter is necessary, use a low value between 1 and 5.

OPTICALREINITMODE

Reinitialization mode for rewritable optical cartridges. Valid values are:

GROUP

Expired rewritable optical cartridges retain original storage group affiliation when reinitialized. This is the default.

OAMSCRATCH

Expired rewritable optical cartridges revert to *SCRTCH* storage group when reinitialized.

The OPTICALREINITMODE keyword is used to determine whether an optical cartridge at reinitialization should maintain its storage group affiliation or revert to a scratch storage group. A cartridge's reinitialization mode is set according to any OPTICALREINITMODE option in effect at the time OSMC Shelf Space Manager selects the cartridge for reinitialization, not according to the options in effect at the time the optical cartridge is physically reinitialized.

To understand how the OPTICALREINITMODE keyword affects the optical volumes at reinitialization, it is important to understand the reinitialization process itself. The following information gives you a overview of the process.

An optical media cartridge contains two logical optical disk volumes, each optical volume is assigned a unique volume serial number (*volser*). The optical media types are either WORM or rewritable.

Shelf Space Manager (a component of OSMC) processes expired optical disk cartridges as follows:
- For WORM cartridges:
  - If all objects on both *volsers* have been deleted and both *volsers* are full and no objects have been written to this cartridge in the past 24 hours, then the cartridge is ejected if it is library-resident. Message CBR2153I is issued to inform the installation that all of the objects on the WORM cartridge were expired and the cartridge was removed from the OAM Optical Configuration Database (OCDB).
- For rewritable cartridges:
  - If all objects on both *volsers* are deleted and no objects were written to this cartridge in the last 24 hours, then the volume empty (VOLEMPTY) indicators in the OAM volume table in the OCDB for both *volsers* contained in the cartridge are set to indicate that the cartridge is ready to be reinitialized. Message CBR2154I is then issued to inform the installation that this rewritable cartridge will be reinitialized the next time it is mounted on an optical drive.
- For all cartridges:
  - The expiration date needs to be the current day or earlier.

Note: The MODIFY OAM,UPDATE,VOLUME command can be used to update the volume expiration date. See "Updating Fields in the DB2 Volume Table and the Tape Volume Table" on page 315 for more information on this command.

When a rewritable optical cartridge that is selected by Shelf Space Manager for reinitialization is mounted, both sides of the cartridge are reformatted. The volumes on the reformatted cartridge retain their original volume serial numbers. With the SETOPT statements, you can specify whether the reinitialized cartridge should maintain its storage group affiliation (default) or revert to the scratch storage group.

SCRENTRYTHRESHOLD
Specifies the amount of free space, in KB, that will determine a WORM optical volume's eligibility to be assigned as a scratch volume. If a new WORM optical volume has less free space than specified, a message is issued to validate the entry or labeling of the volume as a scratch cartridge. The default value is 0.

STORAGEGROUP
Specifies the name of an Object or Object Backup storage group that was previously defined using ISMF. This is the name of the storage group to which the following subparameters apply.

OPTICALREINITMODE

Reinitialization mode for rewritable optical cartridges belonging to this storage group. If this keyword is not specified for a given storage group then the reinitialization mode for rewritable optical cartridges belonging to that storage group will be set per the OPTICALREINITMODE set at the global level.

Note: A cartridge's reinitialization mode is set according to any OPTICALREINITMODE option in effect at the time OSMC Shelf Space Manager selects the cartridge for reinitialization—not according to the options in effect at the time the optical cartridge is physically reinitialized.

Valid values for the OPTICALREINITMODE option are:

GROUP

Expired rewritable optical cartridges belonging to this Object storage group retain their original storage group affiliation when reinitialized.

OAMSCRATCH

Expired rewritable optical cartridges belonging to this Object storage group revert to *SCRTCH* storage group when reinitialized.

SETOSMC Statements for Use in the OSMC Environment

The SETOSMC statement and its associated keywords of the CBROAM*xx* PARMLIB member determine the valid values of settings for OSMC backup processing. They associate an Object storage group with the Object Backup storage groups that store the first or second backup copies of objects. The SETOSMC statement determines which Object Backup storage groups are to contain the first and second copies of the objects that are associated with an Object storage group. If you do not provide any SETOSMC statements, OAM cannot process second backup copies of objects.

The information that is contained in the following topics is provided to allow you to create or update this PARMLIB member for second backup copies of objects support.

Figure 16 on page 110 shows examples of SETOSMC statements in the CBROAM*xx* PARMLIB member that can be used as samples for your installation. The STORAGEGROUP subparameters FIRSTBACKUPGROUP and SECONDBACKUPGROUP can both be specified in a single SETOSMC STORAGEGROUP statement, as shown in the second example in Figure 16 on page 110. The descriptions of the keywords are found in the discussion of the "SETOSMC Keyword Definitions" on page 110.

```
SETOSMC FIRSTBACKUPGROUP(global_1st_bu_group)
        SECONDBACKUPGROUP(global_2nd_bu_group)
        STORAGEGROUP(object_storage_group
                     FIRSTBACKUPGROUP(1st_bu_group))
        STORAGEGROUP(object_storage_group
                     SECONDBACKUPGROUP(2nd_bu_group))

              OR

SETOSMC STORAGEGROUP(object_storage_group
                     FIRSTBACKUPGROUP(1st_bu_group)
                     SECONDBACKUPGROUP(2nd_bu_group))
```

Figure 16. CBROAMxx PARMLIB Member Samples using the SETOSMC Statement and Optional Parameters

Note: The SETOSMC statement is supported at both the global level and at the storage group level. If you specify parameters without an *object_storage_group* name, then OAM will provide the defaults for all Object storage groups in the configuration. If you specify parameters with an *object_storage_group* name, OAM provides the specific Object Backup storage groups to be used for that Object storage group's backup copies.

The syntax for the SETOSMC statement follows.

SETOSMC Statement Syntax: OAM Global Level Parameters

```
►►──SETOSMC──┬──────────────────────────────────────────┬──────────────────►
             └─FIRSTBACKUPGROUP(global_1st_bu_group)─┘

►─┬───────────────────────────────────────────┬────────────────────────────►
  └─SECONDBACKUPGROUP(global_2nd_bu_group)─┘

►─┬──────────────────────────────────────────────────┬─────────────────────►◄
  └─STORAGEGROUP(See STORAGEGROUP Subparameters)─┘
```

SETOSMC Statement Syntax: STORAGEGROUP Subparameters

```
►►──STORAGEGROUP(object_storage_group──┬────────────────────────────────┬──┬──────────────────────────────────┬──)──►◄
                                        └─FIRSTBACKUPGROUP(1st_bu_group)─┘  └─SECONDBACKUPGROUP(2nd_bu_group)─┘
```

SETOSMC Keyword Definitions

The following keywords are defined as they pertain to the CBROAM*xx* PARMLIB member SETOSMC statement:

global_1st_bu_group

This is the default Object Backup storage group that OSMC uses to store the first backup copy of objects when:

- The Object storage group to which the object belongs is not specified in a SETOSMC statement on the FIRSTBACKUPGROUP parameter, and
- The management class that is assigned to the object specifies that a backup copy is to be written.

global_2nd_bu_group
This is the default Object Backup storage group that OSMC uses to store the second backup copy of objects when:

- The Object storage group to which the object belongs is not specified in a SETOSMC statement on the SECONDBACKUPGROUP parameter, and
- The management class that is assigned to the object specifies that more than one backup copy is to be written.

object_storage_group
When a backup copy is to be written for an object which belongs to the Object storage group that is associated with the FIRSTBACKUPGROUP parameter, that write will be directed to a volume that belongs to *1st_bu_group* instead of *global_1st_bu_group*.

When a second backup copy is to be written for an object which belongs to the Object storage group that is associated with the SECONDBACKUPGROUP parameter, that write will be directed to a volume that belongs to *2nd_bu_group* instead of *global_2nd_bu_group*.

1st_bu_group
This is the Object backup group to which the first backup copy of an object belonging to *object_storage_group* is directed when that object is associated with a management class that specifies that a backup copy is to be written.

2nd_bu_group
This is the Object backup group to which the second backup copy of an object belonging to *object_storage_group* is directed when that object is associated with a management class that specifies that more than one backup copy is to be written.

OAMXCF Statements in an OAMplex

In order to support OAM in a Parallel Sysplex environment, in which multiple OAMs share a common DB2 database and communicate between instances of OAM, it is necessary for each instance of OAM that is to be part of an OAMplex to join an XCF group. Creating or updating the CBROAM*xx* PARMLIB member to include the OAMXCF statements is required to assign OAM instances to an XCF group. Each OAM that joins the group must supply the XCF group name and a member name for that instance of OAM in the sysplex. Keywords for the OAMXCF statement for that instance of OAM in the OAMplex are provided in the CBROAM*xx* PARMLIB member that allow specification of the XCF group and member name to become part of the XCF group. All instances of OAM that join the same XCF become an OAMplex. The scope of the OAMplex must match the scope of the DB2 sharing group. See “Using the UPDATE Command to Set OAMXCF Values” on page 314 for information on changing the OAMXCF values dynamically and on defining the values when the CBROAM*xx* PARMLIB member is not used at initialization.

In the Parallel Sysplex, using transaction shipping through XCF, it is possible for a transaction to take too long to complete or for a response to never be returned for completion. OAM fails requests after a certain amount of time in order to free a user from a wait state. The XCFTIMEOUT keyword, available in the OAMXCF statement, allows you to customize time-out values for your environment. These various time-out values (in seconds) specify the length of time that an instance of OAM in the sysplex is to wait for completion of a transaction that was shipped to another instance of OAM in the sysplex. If a response is not received within the specified time-out value, OAM fails the request, returning a non-zero return and reason code to the caller. Many factors determine the expected response time for a transaction, such as optical compared to tape, reads compared to writes, and automated compared to manual environments. OAM provides different time-out values for different transaction types and environments.

Figure 17 is an example of an OAMXCF statement in the CBROAM*xx* PARMLIB member that can be used as a sample in your installation. The descriptions of these keywords are found in the discussion of the "OAMXCF Keyword Definitions".

```
OAMXCF OAMGROUPNAME(OAMGRP1)
       OAMMEMBERNAME(OAMSYS1)
       XCFTIMEOUT(XCFOPTREADA(20) XCFOPTREADM(50)
                  XCFOPTWRITEA(150) XCFOPTWRITEM(150)
                  XCFTAPEREADA(40) XCFTAPEREADM(50))
```

Figure 17. CBROAMxx PARMLIB Member Samples using the OAMXCF Statement and Optional Parameters

The syntax for the OAMXCF statement follows.

OAMXCF Statement Syntax

```
►►──OAMXCF──┬──────────────────────────┬──┬────────────────────────────┬──────────►
            └─OAMGROUPNAME(oamgroup)─┘  └─OAMMEMBERNAME(oammember)─┘

►─XCFTIMEOUT──┬─(──XCFOPTREADA──(seconds)──)──┬──────────────────────────────────►◄
              ├─(──XCFOPTREADM──(seconds)──)──┤
              ├─(──XCFOPTWRITEA──(seconds)──)─┤
              ├─(──XCFOPTWRITEM──(seconds)──)─┤
              ├─(──XCFTAPEREADA──(seconds)──)─┤
              └─(──XCFTAPEREADM──(seconds)──)─┘
```

OAMXCF Keyword Definitions

The following keywords are defined as they pertain to OAMXCF statements:

OAMGROUPNAME
: Identifies the XCF group name that all instances of OAM within this OAMplex are to join. An XCF group name is 1–8 characters long. Valid characters are A–Z, 0–9, and national characters ($, #, @). If OAMGROUPNAME is specified, then OAMMEMBERNAME becomes a required keyword. If OAMXCF statements exist in the CBROAM*xx* PARMLIB member, both OAMMEMBERNAME and OAMGROUPNAME are required.

OAMMEMBERNAME
: Identifies the specific XCF member name that is to be associated with this instance of OAM in the Parallel Sysplex, when this OAM joins the OAM group in the sysplex. An XCF member name is 1–16 characters long. Valid characters are A–Z, 0–9, and national characters ($, #, @). If OAMMEMBERNAME is specified, then OAMGROUPNAME becomes a required keyword. If OAMXCF statements exist in the CBROAM*xx* PARMLIB member, both OAMMEMBERNAME and OAMGROUPNAME are required.

XCFTIMEOUT
: Identifies the number of seconds (1 to 999 999) that this instance of OAM waits for a response that indicates the completion of a shipped transaction from another instance of OAM in an OAMplex.

Note: Seconds=0 indicates that OAM is to wait indefinitely for a shipped transaction completion response from another instance of OAM in the OAMplex. This is the default. The maximum value is 999 999 seconds, which is approximately 11.5 days.

The following are valid time-out values:

(XCFOPTREADA(*seconds***))**
: Indicates the number of seconds that an OAM originating an optical read request, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request.

(XCFOPTREADM(*seconds***))**
: Indicates the number of seconds that an OAM originating an optical read request for a shelf-resident volume, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request.

(XCFOPTWRITEA(*seconds***))**
: Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains real (automated) optical libraries, which is shipped to another OAM within the OAMplex that owns the optical library defined to the object storage group for processing, should wait for completion of the write request.

(XCFOPTWRITEM(*seconds***))**
: Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains pseudo libraries, which is shipped to another OAM within the OAMplex that owns the pseudo library defined to the object storage group for processing, should wait for completion of the write request.

(XCFTAPEREADA(*seconds***))**
: Indicates the number of seconds that an OAM originating a tape read request targeted for an automated tape library dataserver, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request.

(XCFTAPEREADM(*seconds***))**
: Indicates the number of seconds that an OAM originating a tape read request targeted for an MTL, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request.

Updating the PROCLIB

11 ***Update PROCLIB.***

You must perform this step both for migration and at initial installation.

Sample jobs are provided in SAMPLIB to assist you in making the needed additions to PROCLIB. Before running each SAMPLIB member:

- Update the JOB statement.
- Verify that the system name specified is the same as the name of your DB2 subsystem.
- Ensure that the high-level qualifier on the `//OUT DD JCL` statement matches the naming standard at your installation.

11a ***Modify, if necessary, then run CBRIPROC SAMPLIB job.***

You must perform this step for migration at initial installation.

Change and run SAMPLIB member CBRIPROC (see "CBRIPROC" on page 382) to create member OTIS in PROCLIB.

If the DB2 load module library containing DSNALI is not in the LNKLST concatenation, either include the DB2 load module library in the SYS1.LINKLIB concatenation (LNKLST*xx*) or add a STEPLIB DD to this procedure.

If a STEPLIB is used, then that concatenation must be APF-authorized.

11b ***Modify, if necessary, then run CBRAPROC SAMPLIB job.***

Perform this step **only** *if you start the OAM address space for tape library support. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices) and you are not running OSMC for expiration processing.*

If the installation will use OAM object support, modify and run SAMPLIB member CBRAPROC (see "CBRAPROC" on page 382) to create member OAM in PROCLIB.

If you do not modify CBRAPROC, the following member is created as the default:

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,RESTART=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//      'EJECT=&EJECT,RESTART=&RESTART')
//SYSABEND DD SYSOUT=A
```

If you are using object tape support or an OAMplex as your environment, you must update this job step to include 'OAM=&OAM' to provide the necessary parameter for object tape support, and you must supply the default OAM=*xx* (where *xx* is the low order suffix of your CBROAM*xx* PARMLIB member) specification on the PROC statement as the following example indicates:

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,OAM=xx,EJECT=LRW,RESTART=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
//PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//      'OAM=&OAM,EJECT=&EJECT,RESTART=&RESTART')
//SYSABEND DD SYSOUT=A
```

With the PARM=*keyword*, you can specify values for the following parameters:

APLAN
: Specifies the name of the DB2 application plan for LCS. CBROAM is the name of the DB2 application plan specified when an SQL BIND command was issued for the LCS optical configuration database request modules (DBRMs).

EJECT
: This parameter is used to determine which volumes are ejected from an optical library when the library is full and there is a request to add additional volumes to the library. The valid parameter values on this keyword are as follows:

- LRW—least recently written date for the volume. The volumes are ejected based on how long it has been since an object has been written on the volumes, regardless of how often objects are being read from the volume. The volumes which have the oldest

least recently written date are ejected to make room for the additional volumes requested. This is the default parameter.
- LRM—least recently mounted date for the volume. The volumes which are in the slot for the longest time without being mounted are ejected to make room for the additional volumes requested.

MAXS
Specifies the maximum number of OSMC storage management tasks that can be active at one time. The value given for MAXS must not exceed the number of optical drives that is available for storage management processing. If MAXS is not specified, a default of 2 is assigned. See "OAM Cataloged Procedure Parameter (MAXS)" on page 135 for more information about the MAXS parameter.

Note: If concurrent processing includes Object storage groups writing to tape volumes, the correct corresponding (global level) MAXTAPERETRIEVETASKS and MAXTAPESTORETASKS values on the SETOAM statement must be specified. For more information concerning these keywords, see MAXTAPERETRIEVETASKS on page 94, and MAXTAPESTORETASKS on page 94.

OAM
OAM=*xx* specifies the suffix of the CBROAM*xx* PARMLIB member that OAM should process during OAM address space initialization. The two alphanumeric characters (*xx*) must immediately follow the OAM=*keyword* in the PARM field. If the two characters immediately following the OAM=*keyword* are invalid or not specified, error message CBR0025I is issued. OAM only reads PARMLIB member CBROAM*xx* if the OAM=*keyword* is specified on the PARM field of the JCL EXEC statement in the OAM cataloged procedure. If no OAM=*keyword* is specified on the PARM field of the JCL EXEC statement, no PARMLIB member is read by OAM and object tape support is not active. If the object tape support is not active, OAM cannot read any objects back or write any new objects to tape until OAM is initialized with a valid OAM=*xx* specification, and a valid corresponding CBROAM*xx* PARMLIB member. OAM processes PARMLIB member CBROAM*xx* during OAM address space initialization.

OSMC
When YES, OSMC initializes when OAM initializes; when NO, OSMC is prevented from initializing when OAM initializes. (Operator commands requiring OSMC will not execute; the storage management cycle will not run.)

RESTART
This parameter is used to determine if OAM should automatically restart when it receives notification that a new SCDS is activated. The valid parameter values on this keyword are as follows:
- **Yes**—If RESTART=YES is specified, when OAM is notified that an SCDS activation has occurred, OAM automatically restarts. This is the default. How soon OAM is notified of the SCDS activation depends on the time interval that is specified with the INTERVAL keyword in the IGDSMSxx PARMLIB member.
- **No**—If RESTART=NO is specified, when OAM is notified that an SCDS activation has occurred, OAM continues processing normally. Message CBR0092I is issued to acknowledge that a new SCDS has been activated. It is the responsibility of the installation to ensure that the OAM RESTART command be

issued if an OAM address space restart is necessary. See "Restarting the OAM Address Space" on page 309 for details concerning this command.

UNLOAD
This parameter causes the least recently used 3995 optical disk drive, inside an IBM 3995 optical library, to be unloaded after the number of seconds of inactivity specified on the keyword. This unload only occurs if there are no available optical drives within this library. That is, there are no empty online and operational drives. Thus, during periods of inactivity, you can cause at least one drive to be ready to accept the next mount request without first having to do a demount.

The valid parameter values on this keyword are as follows:
- 0 to 9998 specifies the number of seconds of inactivity before the demount may occur.
- 9999 specifies that the cartridge is not to be demounted during periods of inactivity. This is the default.

Note: The UNLOAD keyword applies only to 3995 optical disk drives that are library-resident.

12 ***Verify or create device numbers.***

You must perform this step both for migration and at initial installation. Do **not** *perform this step if you are not using or do not plan to use optical devices.*

Define 3995 device numbers, as well as CTC device numbers, using the hardware configuration definition (HCD). See "Defining 3995 Device Numbers" on page 343 for more information, and refer to *z/OS HCD User's Guide* for more detail on HCD.

Creating DB2 Databases for Object Tables and Directories

Sample jobs for creating databases are provided in SAMPLIB to help install OAM. These sample jobs help you create the DB2 databases, create the application plans, grant authority to use the application plans, and access the databases. Before running the sample jobs, you must:
- Change the JOB statement to meet your installation's requirements.
- Verify that the user ID that is specified on the JOB statement has the correct authority to perform the requested operations.
- Add a JOBLIB for the appropriate DSNLOAD if it is not in the linklist.
- Verify that the subsystem name specified with the SYSTEM keyword on the DSN command is the name of your DB2 subsystem.
- Change the plan name on the RUN statement to match your current DB2 version and release level.
- Start DB2.

If you choose to have SMS manage your DB2 VSAM data sets, create an ACS routine for the VSAM DB2 allocations. Enable the Object and or the Object Backup storage group, enable the volumes, and then validate and activate the SCDS.

Object Databases

Before OAM can operate, you must create object storage databases. These databases contain either objects or information about objects. OAM requires a separate object storage database for each storage group.

Understanding Object Databases:

NOT Programming Interface information

This segment documents information that is provided to help you diagnose OAM problems.

OAM allows multiple DB2 databases to be used for object storage. Each object storage database has an object directory table space, a 4K Object Storage Table space, and a 32K Object Storage Table space. Within each table space is one table (an object directory table, a 4K Object Storage Table, and a 32K Object Storage Table, respectively). Each database has three indexes into the object directory table and one index into each of the object storage tables. The high-level qualifier on the object storage database must match the high-level qualifier on the object storage definition in the SCDS that was created using ISMF and the high-level qualifier and package name in PBIND.

One additional database is used for object storage administration. The object storage administration database name is OAMADMIN. This database contains a management class table space, a storage class table space, and a collection name table space. Within each table space is one table (a management class table, a storage class table, and a collection name table, respectively). The management class table and the storage class table each have one index, and the collection name table has three indexes.

Attention: The information from the OAMADMIN tables is crucial to the operation of OSMC. **IBM strongly advises against altering these tables.**

Each OAM DB2 object storage database has its own separately defined set of VSAM data sets. There is one VSAM data set for each table space, and one VSAM data set for each index within the database.

End of NOT Programming Interface information

Understanding the Database Creation Jobs: Three jobs are supplied in SAMPLIB to assist you in defining the databases to DB2 that are required in your installation (CBRISQL0, CBRISQLX, and CBRISQLY). For more information on these SAMPLIB jobs, see “CBRISQL0” on page 390, “CBRISQLX” on page 394, and “CBRISQLY” on page 397. Three jobs are supplied in SAMPLIB to assist in allocating the VSAM data sets needed for each of the object storage databases you require (CBRIALC0, CBRIALCX, CBRIALCY). For more information on these SAMPLIB jobs, see “CBRIALC0” on page 383, “CBRIALCX” on page 386, and “CBRIALCY” on page 389.

SAMPLIB member CBRISQL0 job allocates and defines the object storage databases. SAMPLIB member CBRIALC0 allocates the data sets needed for each object storage database. There is no minimum number of databases or views that must be created. Additionally, there is no maximum limit to the number of object storage databases you may have in your installation. These jobs can be run multiple times to define as many object storage groups and data sets as are required to suite your environment.

SAMPLIB members CBRIALCX, CBRIALCY, CBRISQLX, and CBRISQLY allocate the data sets and define the databases required for object administration. These four SAMPLIB jobs must be modified and executed successfully before OSR or

OSMC can be used. SAMPLIB members CBRIALCX and CBRIALCY must be run before SAMPLIB members CBRISQLX and CBRISQLY.

Each step number in all the SAMPLIB jobs is unique. Within the jobs that allocate and define the object storage databases, each step number corresponds to the database name qualifier for the data sets being allocated, and to the database name being defined to DB2.

For you to use an OAM object storage database, the allocate job step must be executed, followed by the related DB2 database definition job step. Modify to include the required storage groups and then run the CBRIALCX, CBRIALCY and CBRISQLX, CBRISQLY jobs and the database definition job steps (CBRISQL0) that correspond to the allocation job steps previously run (CBRIALC0). For more information, see step **15** on page 119.

13 ***Add additional steps to the Database Creation Jobs.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After you have identified the databases needed in your installation, locate the SAMPLIB jobs that contain the steps that will allocate the data sets and define the databases. Add steps for each storage group as needed within the jobs. The remaining jobs and steps must be modified before they are executed.

14 ***Modify the OAM Data Set Allocation Jobs.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

All allocation jobs must be modified as follows:
1. Change the JOB statement to meet your installation's requirements.
2. Change *cat_name* to the DB2 catalog name used in your installation.

Individual allocation job steps must be modified as follows:
1. Change *vol_ser* to the volume serial numbers of the volumes where the data sets will reside.
2. Change *pri_alloc sec_alloc* to the number of cylinders (or tracks) needed for the initial size and the secondary extent size for the data set. The entire statement can be changed to TRACKS(*pri_alloc sec_alloc*) if the data set is not expected to be large. This space allocation must be individually determined for each data set needed by the database. Refer to "Estimating Resource Requirements" on page 57 for information about database space allocation calculations.
3. Add allocations for each object storage group defined in your installation.

The allocation job CBRIALCX and CBRIALCY must be modified as above. See, "CBRIALCX" on page 386, and "CBRIALCY" on page 389 for more information.

If you choose to have SMS manage your DB2 VSAM data sets, create an ACS routine for the VSAM DB2 allocations. Enable the storage group, enable the volumes, then validate and activate the SCDS.

15 ***Run the OAM Data Set Allocation Jobs.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After the allocation jobs have been modified, they must be run to successful completion before proceeding. If an error occurs within a particular step, correct the error and rerun either the entire job or just the failing job step.

Run CBRIALC0 for as many groups as needed; then, run CBRIALCX and CBRIALCY. (Because the delete/define refers to specific volume serials, the storage group must be defined with GUARSPACE=YES.) For more information on these SAMPLIB jobs, see "CBRIALC0" on page 383, "CBRIALCX" on page 386, and "CBRIALCY" on page 389.

16 ***Modify the OAM Database Definition Jobs.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

After the data sets have been allocated, run the jobs and steps that define the databases to DB2. The jobs and steps related to data set allocation must be modified before they are executed. Sample jobs are shown in "Appendix B. Sample Library Members" on page 381 and reside in SAMPLIB.

For all database definition jobs for the object storage databases, change the JOB statement to meet your installation's requirements.

All database definition job steps must be modified as follows:
1. Change "DB2" in the statement DSN SYSTEM(DB2) to the subsystem ID for DB2 in your installation.
2. Change the data set name in the statement LIB('DB2.RUNLIB.LOAD') to the data set name used for the DB2.RUNLIB.LOAD data set in your installation.
3. Change *cat_name* to the DB2 catalog name used in your installation. This must be the same name that was used as the *cat_name* in the allocation job.
4. Change the *auth_id* to the IDs authorized to the respective group.
5. Change the PLAN name on the RUN statement to match your current DB2 version and release level.

The database definition jobs CBRISQLX and CBRISQLY must be modified as previously described in the data set allocations job CBRIALCX and CBRIALCY. For OSR or OSMC to run, the CBRISQLX and CBRISQLY jobs must complete successfully. For more information on these SAMPLIB jobs, see "CBRISQLX" on page 394, "CBRISQLY" on page 397, "CBRIALCX" on page 386, and "CBRIALCY" on page 389.

17 ***Run the OAM Database Definition Jobs.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

The DB2 subsystem must be active before starting any of the database definition jobs. All object storage database definition job steps and the object administration database definition job must complete successfully.

Run job CBRISQL0 as many times as needed for the groups used in your installation. Then run jobs CBRISQLX and CBRISQLY. For more information on these SAMPLIB jobs, see “CBRISQLX” on page 394 see “CBRISQL0” on page 390, “CBRISQLY” on page 397, and “CBRISQLY” on page 397.

Creating the Optical Configuration Database

If you plan to start the OAM address space, you must create the optical configuration database. This database contains the Library table, the Drive table, the Slot table, the Volume table, the Deleted Object table, and the Tape Volume table.

Sample jobs are provided to help you create the optical configuration database, create the application plans, and grant authority for the application plans to be used.

18 ***Modify, if necessary, and then run the CBRSAMPL SAMPLIB job.***

You must perform this step at initial installation. For migration, verify that this step was performed during the original OAM installation. If this step was not performed during the original OAM installation, perform this step when you migrate from a previous release. Do **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices).*

Member CBRSAMPL in SAMPLIB (see “CBRSAMPL” on page 399) creates the optical configuration database.

You must make the following changes before running the job:
- In the CREATE STOGROUP statement, add:
  - Your volume serial numbers (VOLUMES parameter).
  - The name of your ICF catalog (use 6 characters in the VCAT parameter).

The following summarizes the changes you can make:
- Choose a name for the DB2 storage group and use that name for each STOGROUP parameter.
- Change a catalog password (PASSWORD parameter) in the CREATE STOGROUP statement.

Browse the output to ensure that the job completed successfully. Each SQL statement executed should have an SQLCODE of 0. Check that the final statement of the sample job (COMMIT) has executed.

Note: The optical configuration database is a single DB2 database with six tables. Exercise caution to avoid creating two optical configuration databases with two sets of tables. Two optical configuration databases can be mistakenly created by running sample job CBRSAMPL twice under two different user IDs. The user ID associated with the job is the user ID supplied by the system programmer with the USER= keyword on the JOB statement.

You may decide to move the data from one media type to another for migration purposes. For example, you may want to move data residing on 12-inch optical media to 5.25-inch optical media to upgrade your installation. This is an option with the use of the Move Volume utility, and is discussed in more depth in “Starting the Move Volume Utility” on page 236. No changes are required to any application program, which currently invokes the OSREQ macro. The only requirement is to use ISMF to define the tape or optical libraries and optical drives to OAM.

19 ***Run the CBRSMB2 SAMPLIB job.***

Do **not** *perform this step at initial installation. You must perform this step when migrating from a previous release (unless the 3590-1 SPE support was already installed) to the current release of DFSMS. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices) or if you are running OSMC for expiration processing.*

CBRSMB2 modifies the Optical Configuration Database by adding a new EPI column to the TAPEVOL table.

20 ***Run the CBRSM150 SAMPLIB job.***

Do **not** *perform this step at initial installation. You must perform this step when you migrate from a previous release to the current release of DFSMS. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices).*

CBRSM150 adds a new MEMBER column in the LIBRARY, DRIVE, VOLUME, and TAPEVOL tables in the optical configuration database. It also adds a new PLIBRARY column to the VOLUME table and primes the field with the value in the LIBRARY table if the volume record is currently shelf-resident.

21 ***Run the CBRSMR13 SAMPLIB job.***

Do **not** *perform this step at initial installation. You must perform this step when you migrate from DFSMS/MVS 1.5.0 or OS/390 V2R10 to the current release of DFSMS.*

The jobs included with the CBRSMR13 SAMPLIB member of PARMLIB modify the object directories. There are ftwo jobs included with this SAMPLIB: SMR13A and SMR13B:

- SMR13A performs the migration from the DFSMS/MVS 1.5.0 or OS/390 v2r10 version of the Optical Configuration Database to the z/OS V1R3 version, which supports the multiple Object Backup storage groups and the second object backup function in OAM/OSMC. This job adds a new column BKTYPE to the existing VOLUME table. It also adds a new column BKTYPE to the existing TAPEVOL table. For recovery purposes, it is recommended that you create a DB2 image copy of the existing VOLUME and TAPEVOL tables prior to executing this migration job.

  See “CBRSMR13” on page 411 for the actual SMR13A SAMPLIB job.
- SMR13B performs the migration from the base version of the OAM Object directory tables to the z/OS V1R3 version, which supports second backup copies of objects. This job adds new columns ODBK2LOC and ODBK2SEC to the existing Object directory tables. For recovery purposes, it is recommended that you create a DB2 image copy of the existing Object directory tables prior to executing this migration job.

  See “CBRSMR13” on page 411 for the actual SMR13B SAMPLIB job.

Note: After you run the CBRSMR13 jobs, you may need to run a DB2 REORG after performing an ALTER to the table.

Merging Object Tables and OCDB for an OAMplex

22 ***Run the CBRSMERG SAMPLIB job.***

Do **not** *perform this step at initial installation. Perform this step when all of the following conditions exist: you are setting up an OAMplex, you currently have multiple OAMs running on separate MVS images in a sysplex, and you want to merge two or more separate OCDBs. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices) or if you are running OSMC for expiration processing.*

CBRSMERG performs a database merge of the OAM configuration databases for use with DB2 data sharing in an OAMplex environment. You may use this sample job or some other DB2 method to perform this database merging. For more information on this SAMPLIB member, see “CBRSMERG” on page 415.

23 ***Run the CBRSG100 SAMPLIB job.***

Do **not** *perform this step at initial installation. Perform this step only when all of the following conditions exist: you are setting up an OAMplex; you currently have multiple OAMs running on separate MVS images in a sysplex; and you want to merge two or more separate OAMADMIN tables, object storage databases, or both.*

CBRSG100 performs a catalog and database merge of OAM databases and catalogs for use with DB2 data sharing in an OAMplex. You may use this sample job or some other DB2 method to perform the catalog and database merging. For more information on this SAMPLIB member, see “CBRSG100” on page 422.

Creating and Binding DB2 Packages

24 ***Run the CBRPBIND SAMPLIB job.***

You must perform this step when migrating from a previous release to the current release of DFSMS. Do **not** *perform this step at initial installation.*

CBRPBIND performs a DB2 BIND of DBRMs to create the packages needed to access the OAM object storage group tables. The use of the DB2 packages allows user defined qualifiers for the object storage groups table definitions. CBRPBIND must be modified and run prior to running the CBRABIND, CBRHBIND, or CBRIBIND jobs. It is currently written for the existing 100 object storage groups with high-level qualifiers of GROUP00–GROUP99, so it needs to be modified for your installation requirements.

OSR Application Plans

For OSR to function correctly, you must create the OSR application plan, bind it to DB2, and grant authority for the plan to be used.

25 ***Run the CBRIBIND SAMPLIB job.***

If you are **not** *planning to create the optical configuration database but you do plan to store objects without starting the OAM address space, you must perform this step both for migration and at initial installation. If you plan to create the optical configuration database, skip this step and proceed to step* **27**.

Creating the OSR Application Plan: You must create the OSR application plan, bind it to DB2, and grant authority for the plan to be used. SAMPLIB members CBRIBIND and CBRIGRNT are provided for this purpose. CBRIBIND binds the packages created in CBRPBIND to the OSR application plan. You need to modify this job to include the high level qualifiers for the installation’s storage groups (currently setup for GROUP00-GROUP99). This step is not required if you plan to create the optical configuration database and start the OAM address space. The

OSR application plan for the optical/tape environment is provided as part of the CBRABIND and CBRAGRNT jobs. After execution, the job output should contain the following message about CBRIDBS:

```
DSNT200I BIND FOR PLAN CBRIDBS SUCCESSFUL
```

If you do not receive this message, refer to "OAM Diagnostic Aids" in the *z/OS DFSMSdfp Diagnosis Reference* manual for more information.

Note: Remember that if OAM has been previously installed, remove the CBRHCICS application plan via the DB2 DSN FREE subcommand for DFSMS 1.2.0.

The DSN FREE subcommand removes information about the CBRHCICS application plan from the following DB2 catalog tables:

SYSIBM.SYSTABAUTH
SYSIBM.SYSDBRM
SYSIBM.SYSPLAN
SYSIBM.SYSPLANAUTH
SYSIBM.SYSPLANDEP
SYSIBM.SYSSTMTDEP

For information on the DSN FREE subcommand, see the *DB2 Command Reference* manual. For more information about the DB2 catalog tables, see the *DB2 SQL Reference* manual.

Note: If you do not plan to use OSMC, refer to the information on the required application plan CBRHSMSI under "OSMC Application Plans."

26 ***Run the CBRIGRNT SAMPLIB job.***

If you are **not** *planning to create the optical configuration database but you do plan to store objects without starting the OAM address space and you have previously run CBRIBIND, you must perform this step both for migration and at initial installation. If you plan to create the optical configuration database, skip this step and proceed to step* **27** *.*

Granting Authority: You must grant applications the authority to use plan CBRIDBS and to access the databases. The statements that grant this authority are shown in SAMPLIB. Also see "Examples of Granting and Revoking Privileges" in *DB2 for OS/390 Administration Guide*.

Completion of a successful grant is indicated by the following message in the job output:

```
DSNT400I SQL CODE=000, SUCCESSFUL EXECUTION
```

If you do not receive this message, refer to "OAM Diagnostic Aids" in *z/OS DFSMSdfp Diagnosis Reference* for more information.

Note: If you have applications containing application plans also using the modules identified in the CBRIBIND SAMPLIB job, you will need to rerun the BIND and GRANT jobs for these *other* applications you have installed as well. The IBM SAA ImagePlus Object Distribution Manager MVS/ESA, the IBM Report Data Archive and Retrieval System (RDARS), and the Item Access Facility

CICS application (IAFC), as well as others are examples of applications that bind OAM and Object Distribution Manager modules into a common plan.

OSMC Application Plans

To use OSMC in your installation, you must create the OSMC application plans, bind them to DB2, and grant authority for those plans to be used. If you are not using OSMC, you must create the application plan for CBRHSMSI, bind it to DB2 and grant authority for it to be used (modify the CBRHBIND and CBRHGRNT sample jobs to include only the statements for CBRHSMSI).

27 ***Run the CBRHBIND SAMPLIB job.***

You must perform this step both for migration and at initial installation.

Creating the OSMC Application Plans: After creating the databases for OSR, you must create the OSMC application plans. Member CBRHBIND in SAMPLIB is provided for this purpose. After running the job, the following DB2 messages should appear in the job output:

```
DSNT200I BIND FOR PLAN CBRHSMSI SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHOBJP SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSOBP SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSVOL SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSBKV SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHRDAS SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHWDAS SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHDUPD SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSBCC SUCCESSFUL
DSNT200I BIND FOR PLAN CBRHSPCC SUCCESSFUL
```

If you do not receive these messages, refer to “OAM Diagnostic Aids” in the *z/OS DFSMSdfp Diagnosis Reference* manual for more information.

Note: The user ID associated with the CBRHBIND job must be the same as the user ID associated with the CBRISQL0 job, because the SQL statements in the OSMC application plans contain unqualified DB2 table names. When an unqualified DB2 table name is encountered during the BIND process, DB2 assumes the unqualified table name is the authorization ID of the binder. For additional information on binding a DB2 application plan, refer to *DB2 Application Programming and SQL Guide*.

28 ***Run the CBRHGRNT SAMPLIB job.***

You must perform this step both for migration and at initial installation.

Granting Authority: To grant authority to use the application plans, run SAMPLIB member CBRHGRNT. There are no changes other than user data that you add to the JCL. If you do not grant this authority, OSMC will not initialize, because it is not authorized to use the application plans.

LCS, ISMF, and OSR Application Plans

After creating the database for LCS, you must create and bind the LCS (CBROAM), ISMF (CBRISMF), and OSR (CBRIDBS) application plans. You must also grant authority for the plans to be used.

29 ***Run the CBRABIND SAMPLIB job.***

If you plan to create the optical configuration database and to start the OAM address space to store objects, you must perform this step for both migration and at initial installation. Do **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices), or if you are running OSMC for expiration processing.*

Creating the LCS, ISMF, and OSR Application Plans: Member CBRABIND binds the packages created in CBRPBIND to the LCS, ISMF, and OSR application plans. Modify this sample job to include the storage group high level qualifiers for the installation (currently set up for GROUP00–GROUP99). This job also includes the addition of the CBRUTIL plan.

After you run the job, the following DB2 messages should appear in the job output:

```
DSNT200I BIND FOR PLAN CBROAM SUCCESSFUL
DSNT200I BIND FOR PLAN CBRISMF SUCCESSFUL
DSNT200I BIND FOR PLAN CBRIDBS SUCCESSFUL
```

If you do not receive these messages, refer to “OAM Diagnostic Aids” in *z/OS DFSMSdfp Diagnosis Reference* for more information.

Note: The user ID associated with the CBRABIND job must be the same as the user ID associated with the CBRSAMPL job, because the SQL statements in the LCS and ISMF application plans contain unqualified DB2 table names. When an unqualified DB2 table name is encountered during the BIND process, DB2 assumes the unqualified table name is the authorization ID of the binder. For additional information on binding a DB2 application plan, refer to *DB2 Application Programming and SQL Guide*.

30 ***Run the CBRAGRNT SAMPLIB job.***

If you plan to create the optical configuration database and to start the OAM address space to store objects and you have created the application plans using CBRABIND in step **29** *, you must perform this step for both migration and at initial installation. Do* **not** *perform this step if you are using DASD-only storage (no optical volumes or tape devices), or if you are running OSMC for expiration processing.*

Granting Authority: To grant authority for these application plans to be used, run member CBRAGRNT. There are no changes other than user data that you add to the JCL. If you do not grant this authority, the OAM address space will not start, because it is not authorized to use the CBROAM application plan, and ISMF and OSR will be unable to gain access to the optical configuration database.

Note: If you have applications containing application plans also using the modules identified in the CBRIBIND SAMPLIB job, you will need to rerun the BIND and GRANT jobs for these *other* applications you have installed as well. The IBM SAA ImagePlus Object Distribution Manager MVS/ESA, the IBM Report Data Archive and Retrieval System (RDARS), and the Item Access Facility CICS application (IAFC), as well as others are examples of applications that bind OAM and Object Distribution Manager modules into a common plan.

Verifying DB2 Installation

After creating all required databases and application plans, and after granting authorization for the entire OAM system (OSR, LCS, and OSMC), do the following verification process:

31 ***Verify that all application plans have been created.***

You must perform this step both for migration and at initial installation.

Use SPUFI to enter the following command:

```
SELECT * FROM SYSIBM.SYSPLAN
  WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

CBROAM	CBRHSVOL	CBRHDUPD	CBRISMF
CBRHSMSI	CBRHSBKV	CBRHSBCC	CBRUTIL
CBRHOBJP	CBRHRDAS	CBRHSPCC	
CBRHSOBP	CBRHWDAS	CBRIDBS	

You must perform this step if you are going to use DASD-only storage (not optical volumes or tape devices).

Use SPUFI to enter the following command:

```
SELECT * FROM SYSIBM.SYSPLAN
  WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

CBRHSMSI
CBRIDBS

Execute the SELECT statement once for each plan. After each execution, one row of information should be returned. Refer to the *DB2 SQL Reference* manual for a detailed description of the information that should be returned.

32 ***Verify that all application plans have been authorized.***

You must perform this step both for migration and at initial installation.

Use SPUFI to enter the following command:

```
SELECT * FROM SYSIBM.SYSPLANAUTH
  WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

CBROAM	CBRHSVOL	CBRHDUPD	CBRISMF
CBRHSMSI	CBRHSBKV	CBRHSBCC	CBRUTIL
CBRHOBJP	CBRHRDAS	CBRHSPCC	
CBRHSOBP	CBRHWDAS	CBRIDBS	

You must perform this step if you are going to use DASD-only storage (not optical volumes or tape devices).

Use SPUFI to enter the following command:

```
SELECT * FROM SYSIBM.SYSPLANAUTH
  WHERE NAME = 'xxxxxxxx';
```

Substitute each of the following plan names for the xxxxxxxx on the WHERE clause:

```
CBRHSMSI
CBRIDBS
```

Execute the SELECT statement once for each plan. After each execution, one row of information should be returned. Refer to the *DB2 SQL Reference* manual for a detailed description of the information that should be returned.

If no information is returned, access authority to the application plan specified on the SELECT statement has not been successfully granted. Refer to "OAM Diagnostic Aids" in the *z/OS DFSMSdfp Diagnosis Reference* manual for directions about how to proceed.

Defining User Catalogs

33 ***Evaluate and implement user catalogs and policies.***

You must perform this step at initial installation. For migration, you must verify or perform this step if you determine that it has not yet been completed in your current environment.

OAM collection names are defined by entries in a user catalog or catalogs. The collection entries in this catalog are made automatically by OAM when the first object is stored to a new collection. Define and maintain the catalogs or catalog aliases using IDCAMS and your installation's standard catalog maintenance procedures. Because access to the collection name entries is essential to OAM operation, ensure that these user catalogs are included in your standard backup and recovery procedures. If you are in an OAMplex, you should have a shared catalog for the collection entries to avoid errors with collections in the DB2 table that are not in the catalog. It is possible to maintain separate catalogs, but that is not the preferred method due to the inconvenience factor.

If no new collections are created, there is no need to share a catalog. It is necessary, in this case, to make sure that all OAMs within the same OAMplex have identical catalog entries in their respective catalogs.

Performance Considerations

For each OSREQ request to OAM by an application program, OAM verifies the collection name supplied by the application. To minimize the time required to verify the collection name, the following recommendation is made:

Shared Catalog — Use unshared catalogs whenever possible to avoid I/O operations to the master and user catalogs. You can use the AMS ALTER command to set catalog SHAREOPTIONS to (3 3), which defines them as unshared. Alternatively, you can place catalogs on unshared volumes. For more information on the AMS ALTER command, refer to *z/OS DFSMS Access Method Services*.

Attention: If you use SHAREOPTIONS option (3 3) and the catalog resides on a shared device, you must ensure that the catalog is accessed from only one system, or unpredictable damage to the catalog may occur.

IPL the System

34 ***IPL the system.***

You must perform this step both for migration and at initial installation.

Use the new I/O configuration definition to IPL the system. The following messages are issued and can be used as verification that the IPL of the system is successful.

```
CBR8001I OAM1 subsystem initialization starting.
```

Note: If your installation is not using OAM for storing objects and is strictly using OAM for tape storage management, the following message can be ignored when it is displayed in response to IPLing the system:

```
CBR8007I No DB2 SSID or the DB2 SSID value of "NONE" has been specified.  OTIS
         subsystem cannot successfully initialize.

CBR8002I OAM1 subsystem initialization completed.
```

Specifying the SMS Definitions and Programs Used by OAM

At this point, OAM is installed; however, you must complete several more steps before applications can be run:

- Define hardware devices, such as optical libraries and optical drives.
- Define OAM-related SMS constructs, such as storage class, management class, storage group, and data class. These constructs express your storage management policy.
- Develop ACS routines to assign constructs to objects. The ACS routines implement your storage management policy.

To complete these tasks, you must first translate your installation's business needs into technical terms. Once that has been accomplished, ISMF must be used to actually create construct definitions and programs in the system.

During installation, customization, and testing, it is reasonable to expect that you will need to adjust the configuration definitions through ISMF. You may also need to perform OAM operator tasks, such as entering optical disks or tape cartridges into associated libraries. Refer to "Chapter 4. Administering OAM" on page 151 and "Chapter 5. Operating OAM and OTIS Address Spaces and OSMC Functions" on page 217 for this information, as well as *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for more information.

Translating the Business Analysis into Technical Definitions

The SMS definitions are the mechanisms by which the results of your business analysis are implemented. The ideal groups, classes, and cycles identified during analysis must now be translated into practical terms. This section discusses the OAM-related SMS parameters and how they can be used to customize OAM for your needs.

As in any translation process, compromises and approximations must be made. For example, you may have to make trade-offs between performance and cost (DASD versus optical versus tape storage) as you implement your storage management policy.

Be alert to the subtle factors that can influence system performance and, therefore, should be considered during this translation process. For example:
- The number of class transitions for a given object during its lifetime can have a significant effect on work load.
- The effective data transfer rate is faster for a single large object than for multiple small objects.

The choice you make for any single parameter has the potential to affect many other parameters. Be prepared to adjust those choices after installation, as you tune and refine your system.

Naming Conventions

Naming conventions are an essential part of managing storage through SMS. For example, the ACS routines use names as part of the processing for evaluating class change requests. Refer to *MVS/ESA SML: Managing Data* for information about establishing and using naming conventions.

Storage Group Parameters

When you define Object or Object Backup storage groups, the following parameters have special significance in an object support environment for optical storage:
- DRIVE STARTUP THRESHOLD
- VOLUME FULL THRESHOLD
- MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION

Also, the following storage group parameters play an important role in controlling the storage management cycle:
- CYCLE START TIME
- CYCLE END TIME
- OSMC PROCESSING SYSTEM NAME

For more information about these parameters, see the information that follows in this section and refer to “Defining Storage Groups” in *z/OS DFSMSdfp Storage Administration Reference*.

DRIVE STARTUP THRESHOLD: This parameter is used to indicate that OAM needs to start another optical drive to handle additional write requests for a specific Object or Object Backup storage group. When the number of requests to write objects to this Object or Object Backup storage group divided by the number of drives currently processing write requests for this storage group exceeds this threshold, OAM starts using an additional optical disk drive for writing to this storage group if one is available.

| **Note:** Tape drive startup thresholds are not determined in the same manner as the thresholds in an optical environment. The drive startup threshold for tape drives is specified in megabytes, while the drive startup threshold for optical volumes is specified by the number of write requests. For information of how the tape drive startup thresholds are determined, see the discussion concerning TAPEDRIVESTARTUP on page 102.

Note that while your planning is typically oriented toward the OSMC storage management cycle, remember that other OSMC functions (for example, Volume Recovery, Move Volume utility, and others) cause objects to be written into specific

Object or Object Backup storage groups as well. If you plan to use these other types of OSMC functions, you need to consider and account for the drives these other types of OSMC functions require in addition to the drives required by the OSMC storage management cycle when determining the appropriate value for the threshold.

Exceeding the DRIVE STARTUP THRESHOLD: Figure 18 is an example of how OAM handles additional write requests for optical when the threshold is exceeded.

```
SCENARIO:

# OF WRITE REQ.   # OF DRIVES           DRIVE STARTUP THRESHOLD
FOR STRGPA        CURRENTLY RUNNING     FOR STRGPA

12                1                     17

During the storage management cycle, more write requests are created
for STRGPA that exceed the DRIVE STARTUP THRESHOLD set for this storage
group.  The following occurs:

# OF WRITE REQ.   # OF DRIVES           DRIVE STARTUP THRESHOLD
FOR STRGPA        CURRENTLY RUNNING     FOR STRGPA

30                2                     17

OAM starts another drive to handle the excess requests for this
storage group as it continues to process requests on the first drive.
```

Figure 18. Another Drive is Started When DRIVE STARTUP THRESHOLD is Exceeded

This parameter allows you to control the number of drives used concurrently for writing objects on optical volumes within a storage group. Setting the DRIVE STARTUP THRESHOLD to the maximum of 9999 with no more than one volume mounted for a storage group increases the likelihood that OAM will write the objects sequentially. The default value for DRIVE STARTUP THRESHOLD is 17.

Improving Performance with Low DRIVE STARTUP THRESHOLD Value: Using a low DRIVE STARTUP THRESHOLD value can allow objects to write concurrently during the storage management cycle, resulting in improved performance and reduced cycle time for the storage group. If more than one drive is used for writing, then the writing of an object to a volume is interspersed with the writing of other objects to their respective volumes. Objects for only one storage group reside on the two optical volumes on an optical disk. Spreading data across volumes makes those volumes unavailable for other storage groups.

You can choose different DRIVE STARTUP THRESHOLD values for different storage groups, allowing you to choose independently between drive write concurrency and the volume-fill characteristic for each of the groups.

The determination of the DRIVE STARTUP THRESHOLD is the same for both Object and Object Backup storage groups.

Note: Make certain that you have enough usable optical disks per storage group (scratch or already assigned to the storage group) to be used simultaneously for the write requests to the storage group once the DRIVE STARTUP THRESHOLD is crossed. OAM does not issue any message to request additional space for a storage group when the DRIVE STARTUP

THRESHOLD is crossed and additional space for the storage group is not available. If the DRIVE STARTUP THRESHOLD is crossed, and there are no usable optical disks available, it is as if the threshold were never crossed; OAM continues to perform the write requests on the existing busy disks. Performance is slower than it would have been if additional space had been available when the threshold was crossed.

Table 21 contains information to help you select a value for DRIVE STARTUP THRESHOLD, based on average object size and the number of drives on which concurrent activity is permitted. If your average object size is slightly larger or smaller than the size shown in the table, adjust the DRIVE STARTUP THRESHOLD value accordingly.

Table 21. Recommended Values for DRIVE STARTUP THRESHOLD

Average Object Size in Bytes	Activity on One Drive Only	Activity on Two Drives	Activity on Three Drives
3 000	>= 4680	2340–4679	1560–2339
4 000	>= 3600	1800–3599	1200–1799
5 000	>= 2880	1440–2879	960–1439
8 000	>= 1800	900–1799	600–899
16 000	>= 864	432–863	288–431
32 000	>= 432	216–431	144–215
40 000	>= 360	180–359	120–179
64 000	>= 216	108–215	72–107
100 000	>= 144	72–143	48–71
128 000	>= 108	54–107	36–53
256 000	>= 48	24–47	16–23
512 000	>= 24	12–23	8–11
1 000 000	>= 9	5–8	3–4
2 000 000	>= 5	3–4	2

VOLUME FULL THRESHOLD: OAM does not select full volumes to satisfy a write request. You may choose to eject full volumes from a library (for example, using ISMF to obtain a list of all full volumes in a library, then using the EJECT line operator) to streamline processing.

Thus, it is important to select a threshold value that allows the volumes to be marked full in a consistent manner. You must consider the size of the objects stored into the group. If the size of the objects is consistent, choose a threshold that is slightly larger than that size. The VOLUME FULL THRESHOLD is set on a storage group basis. When the number of available kilobytes on a volume falls below the VOLUME FULL THRESHOLD for the storage group to which that volume belongs, the volume is marked full and will not be used for any later write requests. If you find that volumes are not being selected for new objects and they are not being marked full, increase the value for this parameter.

Additionally, during OAM initialization, a volume previously set to FULL=Y with the MODIFY OAM,UPDATE command may be subsequently marked not full. This is because the volume’s current tape-full percentage is less than the value of the TAPEPERCENTFULL parameter on the SETOAM statement. If you intend to mark the volumes in an Object or Object Backup storage group as full, then you must also increase the tape volume's percent full value (PFULL on the UPDATE command) to 100.

Note: The combination of the TAPEFULLTHRESHOLD parameter (either at the global or storage group level) and the TAPEPERCENTFULL parameter on the SETOAM statement is used to determine the percent-full-utilization percentage and the tape-full-threshold factor for a tape volume at the storage group level. See the discussions on pages 103 and 104 for more information.

MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION: As an alternative to, or in addition to VOLUME FULL THRESHOLD, consider using the MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION parameter. If the application using the storage group stores objects that have a wide variety of sizes, use this parameter so the volumes are marked full as determined by the ability of the volume to contain the object. If there is insufficient space on the volume for the object, choosing this parameter causes a volume to be marked full and an alternate volume to be chosen. For example, a request to write an object larger than 30K causes the demount of a volume with a full threshold of 20K if there is less than 30K freespace, in preference to a volume that contains sufficient space. If only the VOLUME FULL THRESHOLD parameter is used, the volume is not marked full; if the MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION parameter is used, the volume is marked full.

An additional side effect of not marking volumes full is the potential for additional volume mounts. A volume is selected according to its ability to contain the next object written with the intent of filling that volume. If there is a small amount of freespace, the volume may be mounted to write one object. If the next object does not fit on that volume, it is demounted to mount another volume that is capable of containing the next object.

CYCLE START TIME and CYCLE END TIME: As defined on page 51, the purpose of the storage management cycle is to ensure that every object scheduled for processing is placed in the correct level of the object storage hierarchy, is deleted or backed up, and, if necessary, is flagged for action during a later storage management cycle. There are five methods by which management cycles can be controlled:

- **Manual Start (All Groups)**

  The storage management cycle can be started manually for all storage groups by using the OAM START version of the MVS MODIFY command (see "Starting OSMC Functions" on page 223 for command syntax). START OSMC starts all groups that either have the current system or no system name specified as the OSMC processing system.
- **Manual Start (Individual Group)**

  The storage management cycle can be started manually for an individual storage group by using the STORGRP keyword on the OAM START version of the MVS MODIFY command. Using this approach, you can directly control the processing sequence, such as using this technique to give priority to a storage group with many objects.
- **Manual Stop**

  The storage management cycle can be stopped manually by using the OAM STOP version of the MVS MODIFY command (see "Stopping OSMC" on page 319 for command syntax).

- **Manual Stop (Individual Group)**

  The storage management cycle can be stopped manually for an individual Object storage group by using the STORGRP keyword on the OAM STOP version of the MVS modify command.
- **Cycle Start Window**

  The storage management cycle can be started automatically by specifying CYCLE START TIME and CYCLE END TIME parameters in the storage group definitions. If using the automatic startup in an OAMplex, you should specify an OSMC processing system name to avoid multiple starts for the same group on different systems. If OSMC is started during the window delimited by those times, the storage management cycle will be started for that storage group. This is the usual method for controlling storage management cycles. You can also indicate that no automatic processing for the storage group is to be performed by specifying NONE for the CYCLE START TIME and leaving the CYCLE END TIME blank.

All Object storage group definitions must define a window where the storage management cycle starts for the storage group or indicates that no automatic processing be performed for the storage group. Consider the following issues as you select window start and end times for each storage group:
- Number of optical drives
- Number of tape drives
- Number and size of objects moving through the hierarchy
- Backup requirements
- Time required to process the group
- Impact on end users who might be doing retrievals
- Storage management cycles for other groups
- Application usage patterns
- General maintenance operation requirements

Note: CYCLE START TIME and CYCLE END TIME describe a window during which the storage management cycle may *start*; they do *not* define the length of the processing period, as that is determined by the duration of time it takes for the storage management cycle to complete. The storage management cycle may still continue to run after the specified CYCLE END TIME has passed.

Processing during the storage management cycle for a group will not require use of an optical or tape drive under the following conditions:
- The storage group does not specify a library.
- There is no class transition that requires moving an object to optical or tape storage.
- No objects require backup.

Storage management cycle processing requires at least one drive if any objects are moved or backed up to optical or tape storage (see "DRIVE STARTUP THRESHOLD" on page 129 and *TAPEDRIVESTARTUP* on page 102 for other considerations). If a storage management cycle is in process on more than one Object storage group at a time and the number of groups exceeds drive availability, frequent volume mounts occur in an attempt to satisfy the requests to write objects to optical and or tape volumes. For example, when objects are written to a mounted volume for one group, that volume must be demounted to allow the mounting of another volume to accept the objects for a different group. Unless you limit resource consumption during the storage management cycle by some other means (see

“OAM Cataloged Procedure Parameter (MAXS)” on page 135), you must not specify overlapping start windows for more groups than you have drives.

If an object requires more than one backup copy, the first and second backup copies are written to separate Object Backup storage groups. Objects are written to backup volumes in the Object Backup storage group specified in the SETOSMC statement for the Object storage group to which they belong, or, if a SETOSMC statement is not specified, the backup copies are written to backup volumes in the default Object Backup storage group specified.

More than one Object storage group can use the same Object Backup storage group for the same backup copy (first or second); therefore, first backup copies of objects from one group can reside on the same volume as first backup copies from other groups and second backup copies of objects from one group can reside on the same volume as second backup copies from other groups. These objects are written during the storage management cycle for the group containing the object. If some groups require object backup, but some do not, consider processing groups that require object backup concurrently with groups that do not require object backup.

It is recommended that you process the storage management cycle for an Object storage group while other activity for the objects in the Object storage group is light. For example, specify a cycle start window during a period when applications are not accessing data heavily. You must consider the effect of concurrent object use in a group during the storage management cycle for that group. DB2 performs deadlock detection on tables (directory tables in particular) that are shared by tasks performing the storage management cycle processing and by user tasks requesting OAM functions through the application interface. The potential for DB2 deadlocks is much greater if an application is accessing data in a group during the storage management cycle for that group.

The cycle time specified is only a start window; it does not define the end of the storage management cycle. You should consider adjusting the start time or times if the processing for one group extends into the start time for another; otherwise, resource contention could become severe enough to affect the total time it takes to perform storage management cycle processing for all groups.

OSMC Processing System Name

The OSMC processing system name specifies the system where the OSMC processing for an object storage group is to be performed. This name is used within an OAMplex where there are multiple instances of OAM running on separate MVS images in a Parallel Sysplex sharing a common configuration and DB2 database. When the OSMC cycle window occurs, the system specified in the OSMC processing system name of the object storage group is where the storage group automatically starts.

If you use the F OAM,START,OSMC command to start a full OSMC cycle, the storage groups with an OSMC processing system name that matched the system where the command was entered will be started. Also, storage groups with no specified OSMC processing system name will be started.

If using the F OAM,START,STORGRP command, the storage group specified will be started on the system when the command was entered. Any specification in the OSMC processing system name for that storage group is ignored.

It is recommended that the system name specified for a storage group be the same system where the hardware associated with the object storage group is connected. If the object storage group has libraries associated with it that are owned by different OAMs in the OAMplex, then determine if there is a higher volume of data localized to one instance of OAM and use that system as the OSMC processing system name. This may need to be modified and tuned to meet performance objectives for the installation.

OAM Cataloged Procedure Parameter (MAXS)

The storage management cycle start time in the storage group definition is the normal method for distributing resource consumption for completion of the storage management cycle. If cycles for groups overlap (either planned or unexpected, because processing did not end when anticipated), MAXS can be used to limit resource consumption. The MAXS parameter in the OAM cataloged procedure (see the discussion on page 114) specifies the number of Object storage groups that can be processed concurrently during a storage management cycle.

In planning for MAXS, you also must consider that OSMC functions other than the storage management cycle (for example, volume recovery utility, move volume utility, and others) are consumers of resources as well, but are not controlled by MAXS. If you plan to use these other types of OSMC functions, you need to consider and account for the resources these other types of OSMC functions require first and then distribute the remaining resources for the OSMC storage management cycle with MAXS.

Optical and tape device availability are the resources most likely to cause contention. Writing active data to optical media during the storage management cycle is done on an Object storage group boundary. No active data is mixed between groups; therefore, each group being processed requires a different optical volume. Concurrent requests for different volumes are likely to result in concurrent requests for optical drives. For example, if you have a single, four-drive library and there are concurrent storage management cycles for four storage groups requiring the writing of data to optical media, all four drives are used. If there are concurrent requests to retrieve data from optical volumes during the cycle, those requests and the processing performed during the storage management cycle contend for resources and detract from the performance of each of the functions. Consider using the MAXS parameter with DRIVE STARTUP THRESHOLD to limit resource consumption for writing objects to optical media during storage management cycles, thereby leaving resources available to mount volumes for retrieval requests.

For object tape support, the limits set in the parameters and subparameters of the SETOAM statement for TAPEDRIVESTARTUP, MAXTAPESTORETASKS, and MAXTAPERETRIEVETASKS limit resource contention regarding tape library dataservers. These parameters can work in conjunction with the MAXS parameter resulting in effective resource utilization of the entire storage management environment. For more information on these and other parameters associated with the SETOAM statement, see the discussion on page 91.

The default value for MAXS is two. This default was chosen as a reasonable value when the configuration includes one library with four optical drives and two stand-alone drives. This default allows for concurrent storage management processing for two storage groups and it also allows for overlapping the writing of backup copies to stand-alone drives with the writing of primary optical copies to the library drives. Also, it leaves spare library drives available for retrieve requests and as alternate drives in case of a drive failure.

MAXS could possibly be increased to four when a second library is added to the configuration. Do *not* set MAXS to a value larger than the number of optical or tape drives that are available for storage management processing. Before increasing the value of MAXS, you should verify that there is sufficient processing capacity available to support the increased work load, because processing requirements are heavier for small objects than for large ones. Also, you need to consider the tuning guidance described in "Tuning OAM" on page 170.

In addition, it is necessary to assign Object and Object Backup storage groups across the libraries in such a way that the library work loads are balanced. This storage assignment prevents any library from becoming a bottleneck.

Storage Class and Management Class Parameters

The OAM-related parameters for defining storage and management classes are described in "Defining Storage Classes" on page 142 and "Defining Management Classes" on page 143. The parameters are not inherently difficult to understand; however, implementing them effectively can be challenging.

During the translation process, establish parameter values for each class; then, evaluate the entire set of classes to ensure that the performance, retention, backup, and processing cycles they define correspond to the requirements established during the business analysis phase.

ACS Routine Input Variables

Automatic class selection routines are used to implement your installation's storage management policy. These routines must be written using the ACS programming language, a high-level language that uses relational statements to determine class and storage group assignments. The *z/OS DFSMSdfp Storage Administration Reference* manual contains detailed information about the ACS programming language and the use of ISMF to define and to validate ACS routines.

The remainder of this segment discusses variables applying specifically to object processing.

The ACS routines use three values for the &ACSENVIR variable that are specific to objects. See Table 22 on page 137 and Table 23 on page 138 for a diagram of these variables. Using these values, you can distinguish object selection from data set selection. These values are as follows:

&ACSENVIR='STORE'

The storage class, management class, and perhaps the storage group routines are invoked because of an application's request to store an object. Variable &DSN contains the collection name.

If variable &MEMN (*object name*) is null, the ACS routines are invoked to specify a storage class, management class, storage group for the collection named in &DSN. Therefore, you must supply ACS routines that select a storage class, management class, and Object storage group for the collection. The storage class and management class supplied by the ACS routines become the default classes for the collection. The storage group selected indicates in which Object storage group the collection is to be a member. All objects in the collection are stored in the storage group that you select.

If variable &MEMN is not null, this ACS invocation validates the storage class and management class specified by the application for the object named in variable &MEMN. ACS routines ensure that the stated class is acceptable for use with this object, and if not, should substitute an acceptable one. This does not affect the class specifications for the collection.

When neither storage class nor the management class is specified on a request to store an object into an existing collection, the object is assigned the default classes associated with that collection.

Attention: When an object is assigned the default classes associated with the collection, the ACS routines are *not* invoked, and it is possible to store an object with a name that does not conform to the requirements in the ACS CHANGE or CTRANS environments. *Subsequent attempts to process that object will fail*.

&ACSENVIR='CHANGE'

ACS is invoked to validate an application's request to change the storage class or management class for the object named in variable &MEMN that is part of the collection named in variable &DSN. ACS routines should ensure that the stated class is acceptable for use. The appropriate ACS routine is invoked based on the combination of storage class (SC) and management class (MC) specifications included on the application request:

- If only MC is specified, the management class ACS routine is invoked, using the requested MC and the existing SC.
- If only SC is specified, both the storage class and management class ACS routines are invoked, using the requested SC and the existing MC.
- If MC and SC are specified, management class and storage class ACS routines are invoked, using the requested MC and SC.

&ACSENVIR='CTRANS'

During a storage management cycle, ACS is invoked because a class transition event has occurred for the object named in variable &MEMN that is part of the collection named in variable &DSN. Variables &MGMTCLAS, and &STORCLAS have the names of the classes to which the object is currently assigned. ACS routines should select the new classes. These new classes can change the placement of the object in the hierarchy and can change the management of the object (including creation of a new transition event).

Table 22. Constructs Verified or Changed through ACS Routines Invoked by the &ACSENVIR Variables—All OAM Environments

ACS ROUTINES INVOKED (&ACSENVIR)	**STORE**	**CHANGE**	**CTRANS**	***ALLOC**
DATA CLASS	NO	NO	NO	YES
STORAGE CLASS	**YES	***YES	YES	YES
MANAGEMENT CLASS	**YES	***YES	YES	NO
STORAGE GROUP	**YES	NO	NO	YES

Note:

*ACS environment of ALLOC is invoked by MVS during allocation.

**When storing the first object into a new *collection* through the STORE request, the ACS routines for storage class, management class, and perhaps the storage group are entered.

The ACS routines for management class and storage class are entered for *objects* only when an explicit management class, storage class, or both are specified on a CHANGE request. (See the ***CHANGE Environment Only graphic for more information.)

Table 23. Constructs Verified or Changed through ACS Routines Invoked by the &ACSENVIR Variables—Change Environment Only

ACS ROUTINES INVOKED (OSREQ SPECIFIES)	STORAGE CLASS ONLY	MANAGEMENT CLASS ONLY	STORAGE CLASS AND MANAGEMENT CLASS
STORAGE CLASS CHANGED	YES	NO	YES
MANAGEMENT CLASS CHANGED	NO	YES	YES

In addition to &ACSENVIR, ACS routines may also use the following:

- Read-Write Variables
  &MGMTCLAS
  &STORGRP
  &STORCLAS
  &SYSNAME
  &SYSPLEX
- Read-Only Variables
  &DSN
  &HLQ
  &LLQ
  &MEMHLQ
  &MEMLLQ
  &MEMN
  &MEMNQUAL
  &NQUAL
  &RETPD
  &SIZE

Note: &SIZE is valid **only** for STORE requests. &SIZE contains the object size converted to kilobytes (KB) and rounded up to the next highest 1 KB if the object size is not at a KB boundary (KB=1 024 bytes).

For detailed information about ACS variables, refer to the “Read-Write Variables” and “Read-Only Variables” sections in the “ACS Language Reference” chapter of the *z/OS DFSMSdfp Storage Administration Reference*.

Storing Objects in a Collection

When the first object is stored in a collection with an OSREQ STORE, the Object storage group for that collection is derived by the SMS Storage Group ACS routine, and the collection name catalog entry is defined with the default management class and storage class. After an object is stored, the collection name entry in the catalog is used to locate the object (see Figure 7 on page 24 for a diagram of the process of storing an object). The catalog entry for the collection name contains a directory token that is used to determine which DB2 database contains the object directory entry for the object. If a collection name entry from the collection name table is lost, objects in that collection will not be processed by the storage management cycle.

Creating OAM Definitions with ISMF

After the translation process has been completed, it is necessary to define all the OAM elements to the system. ISMF provides a series of panels through which SMS parameters can be defined. Refer to *z/OS DFSMS: Using the Interactive Storage Management Facility* for detailed information about using ISMF. You may also want to read “Appendix A. Sample Optical Hardware Configurations” on page 323 through

"Optical Disk Drives" on page 326 before you begin using ISMF. This will familiarize you with hardware configuration issues and explain the CTC / 3995 device numbers needed for the library. For ISMF information about tape volumes and libraries, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Grant DB2 authority to the OAM configuration databases to the ISMF user who will be creating the optical constructs.

Defining an SCDS

35 ***Define the base SCDS.***

You must perform this step at initial installation. During migration, you may optionally perform this step if you are creating additional SCDSs.

The procedure for defining a source control data set is provided in the *z/OS DFSMSdfp Storage Administration Reference*. It is possible to define several source control data sets describing different configurations; however, only one SCDS can be activated at any time.

36 ***Define libraries and drives in the optical configuration database.***

| *You must perform this step at initial installation if you are using optical storage in
| your hierarchy. During migration, you may optionally perform this step if you are
| adding or changing libraries or drives.*

Defining optical libraries and optical drives in the optical configuration database results in data being entered in the DB2 tables. It is therefore necessary that the TSO ID of the user entering the ISMF definitions has access to DB2. For a more comprehensive discussion of defining optical libraries and optical disk drives or for other topics such as deleting, altering, or copying optical libraries or drives, see "Appendix A. Sample Optical Hardware Configurations" on page 323. For ISMF information regarding tape libraries, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Defining Storage Groups and Relating the Libraries to the Storage Groups

37 ***Define storage groups.***

You must perform this step at initial installation. During migration, you may perform this step if you are adding or changing storage groups, or adding libraries to your configuration.

Use the Storage Group Application Selection panel to specify one of the following storage group types:

STORAGE GROUP TYPE	DESCRIPTION
OBJECT	Contains primary objects.
OBJECT BACKUP	Contains backup copies of objects.

Use the Object Storage Group Define panel to specify the following:

DESCRIPTION
Provide a free-form description of this storage group (up to 120 characters).

QUALIFIER
For Object storage groups, specify the name of the DB2 object storage

database to be used for the object directory and the DASD high level qualifier of the storage group. This qualifier must also be defined as a package in the CBRPBIND job to create the package. This package must also be in the CBRHBIND and CBRABIND or CBRIBIND jobs to bind the package to the OSMC, OAM, and OSR plans.

CYCLE START TIME
For Object storage groups, specify the beginning of a window of time when OSMC can begin daily processing for this storage group. You must specify a value of from 0 to 23, or NONE for all Object storage group definitions. A value of from 0 to 23 represents an hour of the day. Specify an hour of the day as 00 for midnight, 01 for 1 A.M., 23 for 11 P.M., and so on. The hour of the day value for CYCLE START TIME must be different from the hour of the day value for CYCLE END TIME. A value of NONE indicates that no automatic processing for the storage group be performed. When NONE is specified, the CYCLE END TIME value must be blank. See "CYCLE START TIME and CYCLE END TIME" on page 132 for more information about this parameter.

CYCLE END TIME
For Object storage groups, specify the end of a window of time when OSMC can begin daily processing for this storage group. You must specify a value from 0 to 23, or blank (depending on your specification for CYCLE START TIME) for all Object storage group definitions. A value of from 0 to 23 represents an hour of the day and is required when a value from 0 to 23 was specified for CYCLE START TIME. Specify an hour of the day as 00 for midnight, 01 for 1 A.M., 23 for 11 P.M., and so on. The hour of the day value for CYCLE END TIME must be different from the hour of the day value for CYCLE START TIME. A value of blank is required when NONE has been specified for CYCLE START TIME. See "CYCLE START TIME and CYCLE END TIME" on page 132 for more information about this parameter.

LIBRARY NAMES
Specify the names of the optical disk libraries in your configuration (either one to eight pseudo or real optical libraries) that can contain volumes belonging to this Object or Object Backup storage group to which objects are written.

VOLUME FULL THRESHOLD
For Object or Object Backup storage groups, specify the number of free kilobytes to be used as a threshold for optical volumes that belong to this storage group. When the number of free kilobytes falls below this threshold, the volume is marked full and no more objects are placed on the volume. See "VOLUME FULL THRESHOLD" on page 131 for more information about this parameter.

If the optical volume table of contents is full, the volume is marked full regardless of what is specified in this parameter.

DRIVE STARTUP THRESHOLD
For Object or Object Backup storage groups, specify the point at which OAM starts using an additional optical disk drive for writing. When the number of requests to write objects to this Object or Object Backup storage group divided by the number of optical drives currently processing write requests for this storage group exceeds this threshold, OAM starts using an additional optical disk drive for writing. The default value for DRIVE STARTUP THRESHOLD is 17. See "DRIVE STARTUP THRESHOLD" on page 129

page 129 for more information about this parameter, including a table of recommended values, based on average object size.

OSMC PROCESSING SYSTEM NAME
For Object storage groups, specify on which system OSMC processing is to run when this storage group is automatically started during the OSMC cycle window specified in the Object storage group definition. Specifying an OSMC processing system name avoids contention by preventing the storage group from being started automatically on multiple systems concurrently within an OAMplex.

MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION
For Object or Object Backup storage groups, specify whether the volume should be marked full at first write-failure.

If you specify **Y** (YES), OAM marks full an optical or tape volume in this Object or Object Backup storage group the first time an object cannot be written on this volume because there is not enough space remaining on the volume.

If you specify **N** (NO), OAM marks an optical volume in this Object or Object Backup storage group full only when the number of available kilobytes in the user data area falls below the VOLUME FULL THRESHOLD.

If the optical volume table of contents is full, the volume is marked full regardless of what is specified in this parameter.

If you specify **N** (NO), OAM marks full a tape volume in this Object or Object Backup storage group only when the number of available kilobytes falls below the TAPEFULLTHRESHOLD that is specified on the SETOAM statement in CBROAM*xx* member of PARMLIB.

See "MARK VOLUME FULL AT FIRST WRITE-FAILURE OPTION" on page 132 for more information about this parameter.

SMS STORAGE GROUP STATUS
For Object or Object Backup storage groups, specify **Y** (YES) for this parameter. This will display another panel where you can specify one of the following values on the SMS Storage Group Status panel for the system running OAM if it is not in an OAMplex, or for the systems in an OAMplex:

ENABLE Applications can store and retrieve objects in the group; OSMC can process the group.

DISNEW Applications can retrieve objects from the group, but cannot store objects into the group; OSMC can process the group.

DISALL Applications can neither store nor retrieve objects in the group; OSMC can process the group.

NOTCON Neither applications nor OSMC can process objects in the group.

Attention: In a *nonOAMplex* environment, use the SMS Storage Group Status panel to allow status to be specified for each system defined to SMS. You should specify an option **other than** NOTCON for the one system that will be running OAM, and you must specify the NOTCON option for all other systems. If you specify more than one system as other than NOTCON for an Object or Object Backup storage group, a message is issued during OAM initialization and the storage group is ignored by OAM.

Also, storage group enablement status should not be confused with library connectivity. Changing the connectivity of a library does not automatically change the enablement status of any associated storage group. Storage group definitions may need to be updated to provide the correct storage group enablement status should the library connectivity be changed.

Defining Storage Groups to Direct Data to Specific Optical Media Types

With the IBM 3995 multifunction optical disk drives, a customer can choose to populate a single library with both WORM and rewritable optical media. This allows the customer to direct data for a particular application to the WORM media while directing data for another application to the rewritable media on the same optical device.

To achieve this control over which media type is used for a specific application's data (objects) residing in the same optical library dataserver as another application's data, it is recommended that the following steps be completed:

- Define one or more Object storage groups for the application that wishes to have its objects stored on WORM media and one or more Object storage groups for the application that wishes to have its objects stored on rewritable media. Also include the library name associated with the library model in the Object storage group definition for each application.
- Enter the media into the 3995 optical library dataserver. The optical disk volumes residing on the WORM or rewritable media should be associated with one of the Object storage groups set up for the application that requires its data to be stored on the specific media type.
- Update the SMS storage group ACS routine to insure that the data belonging to the individual applications is assigned to the appropriate Object storage groups that only have WORM or rewritable optical disk volumes associated with them.
- Make certain there are sufficient optical disk volumes with available space and of the appropriate media type assigned to each Object storage group at all times.
- Make certain there are NO OAM scratch optical disk volumes in the multifunction 3995 Optical Library Dataserver. An OAM scratch optical disk volume can be assigned by OAM to any Object storage group when the Object storage group encounters an 'out-of-space' condition. All optical disk volumes should be preassigned to a given Object storage group based on the media type of the optical disk volume.

Another way for a customer to direct data belonging to different applications to different types of media is:

- Having two or more multifunction 3995 Optical Library Dataservers and populate one with WORM media and the other with rewritable media.
- Defining one or more Object storage groups for the application or applications for which you want data stored on either WORM or rewritable media, and including the library name associated with the multifunction library in the Object storage group definition.
- Using the Default Media Type option to restrict the media type that can be entered into a specific optical library dataserver. See the discussion on page 356 for more information on the Default Media Type option.

Defining Storage Classes

38 ***Define storage classes.***

You must perform this step at initial installation. During migration, you must perform this step if you are adding or modifying storage classes for object tape support.

OAM interprets the parameters used to define the storage class in an attempt to apply the stated performance objective. The following parameters are used by OAM as an indication of the performance objective for the object:

INITIAL ACCESS RESPONSE SECONDS (IARS)
Specify a performance objective relative to the elapsed time (in seconds) that can be tolerated before the first byte of data is made available for an application's request to retrieve an object. Use from 1 to 4 characters to specify a valid value of 0 to 9999. A value of 0 causes the object to be written to DASD storage, and a value of greater than 0 causes the object to be written to removable media storage. Any OAM request that tries to use a storage class with a blank value for this parameter will fail.

SUSTAINED DATA RATE (SDR)
A subparameter of the storage class parameter that specifies which removable media, optical or tape, is used to accept the primary copy of the object, once the Initial Access Response Seconds parameter determines that the object should be written to removable media. If the SDR is greater than or equal to three, the primary copy of the object is stored on a tape volume. If the SDR for the object is less than three, the primary copy of the object is stored on an optical disk volume.

OAM attempts to meet the performance objective by placing the object at a level in the storage hierarchy that comes closest to the objective. Avoid using performance objectives that force objects to be written directly to optical storage. Writing objects directly to optical media without staging them through DASD can degrade system performance and significantly increase the number of optical disks needed per day due to inefficient optical VTOC directory space utilization. See Table 13 on page 64 for detailed information about the effects of writing objects directly to optical media.

AVAILABILITY
Specify a value for this parameter (STANDARD or CONTINUOUS), even though it is ignored for objects.

Defining Data Classes

39 ***Define data classes.***

You must perform this step if you are using object to tape support in your environment and want to direct your tape related Object writes to an ATLDS or an MTL.

You need to specify a DATACLASS to be used to direct work requests to an ATLDS or an MTL. This DATACLASS must be added to the CBROAM*xx* PARMLIB member for each storage group that is directing Object tape writes to the ATLDS or MTL. See "Understanding the Data Class Construct" on page 22 for more details concerning data classes and refer to *z/OS DFSMSdfp Storage Administration Reference* for details on how to define your data classes for your installation.

Defining Management Classes

40 ***Define management classes.***

You must perform this step at initial installation. During migration, you may optionally perform this step if you are adding management classes.

Pages 1 through 3 of the Management Class Define panel are primarily for DFSMShsm's management of data sets. OAM uses the following subset of those parameters to manage objects in the same manner that DFSMShsm manages data sets. Refer to “Defining Management Classes” in *z/OS DFSMSdfp Storage Administration Reference* for more information.

EXPIRATION ATTRIBUTES

Specify when an object can be deleted automatically by OAM, if approved by the auto-delete installation exit. Refer to “Auto-Delete Installation Exit (CBRHADUX)” on page 509 and *z/OS DFSMS Installation Exits* for information about the installation exit.

Note: A value of NOLIMIT for the EXPIRATION ATTRIBUTES means OAM will not automatically delete the objects that are associated with this management class. Those objects must be explicitly deleted by the application, using the application interface to OAM. Refer to *z/OS DFSMSdfp Storage Administration Reference* for a detailed discussion of the relationships between the EXPIRATION ATTRIBUTES and the RETENTION LIMIT values.

EXPIRE AFTER DAYS NONUSAGE

Specify when the object is to be automatically deleted by OAM relative to the elapsed time since it was last referenced, or if the object has not been referenced (such as the case of an object stored today, because its last reference date is set to today’s date), the elapsed time since it was created.

Note: Do not use UPD=**N** on the OAM1 statement in the IEFSSN*xx* member of PARMLIB if this option is used in your management class.

EXPIRE AFTER DATE/DAYS

Specify when the object is to be automatically deleted by OAM, relative to the elapsed time since it was created or on an explicit date.

RETENTION LIMIT

Specify the retention limit allowed on explicit parameters on the application interface to OAM.

AUTO BACKUP

Specify whether you want the object to be backed up by writing one or two copies of the object. The backup copies are made during the OSMC storage management cycle. If you set AUTO BACKUP=**Y** and the number of backup copies specified in the number of backup versions field is zero or one, and you have not specified a SECONDBACKUPGROUP keyword on any SETOSMC statement in the CBROAM*xx* member of PARMLIB, then OSMC schedules a single backup copy of the object to be written. If you set AUTO BACKUP=**Y** and the number of backup copies that is specified in the number of backup versions field is >1, and you have specified a SECONDBACKUPGROUP keyword on any SETOSMC statement in the CBROAM*xx* member of PARMLIB, then OSMC schedules two backup copies of the object to be written.

NUMBER OF BACKUP VERSIONS

Specify the number of backup versions to be made for an object when OSMC processing is done for an Object storage group.

Valid values for the number of backup versions field are as follows:

- 0—creates one backup copy

Note: When AUTO BACKUP=Y in the management class construct, ISMF/SMS will not accept a value of “0” for the number of backup versions.

- 1—creates one backup copy
- ≥2—creates two backup copies

Page 4 of the Management Class Define panel is not used for data sets; it is used for objects to define an event that causes OAM to invoke ACS for the purpose of selecting a new storage class or management class. These class transition events are:

TIME SINCE CREATION
Specify the time (YEARS, MONTHS, and DAYS) that must elapse relative to the date the object was created. The YEARS, MONTHS, and DAYS attributes can be used separately or in combination. A maximum date of 9999/12/31 will be used if TIME SINCE CREATION results in a date exceeding the maximum.

TIME SINCE LAST USE
Specify the time (YEARS, MONTHS, and DAYS) that must elapse relative to the date the object was last referenced, or if the object has not been referenced (such as the case of an object stored today, because its last reference date is set to today’s date), the elapsed time since it was created is used. The YEARS, MONTHS, and DAYS attributes can be used separately or in combination. A maximum date of 9999/12/31 will be used if TIME SINCE LAST USE results in a date exceeding the maximum.

Note: Do not use UPD=N on the OAM1 statement in the IEFSSN*xx* member of PARMLIB if this option is used in your management class.

PERIODIC
Specify that a class transition will occur at a regular period, based on the calendar (regardless of when the object was created or referenced). This parameter has five attributes, which can be used either separately or in combination:

MONTHLY ON DAY
Specify FIRST, LAST, or a number from 1 to 31 indicating the day of the month on which class transition should occur; leave blank if unused.

QUARTERLY ON DAY
Specify FIRST, LAST, or a number from 1 to 92 indicating the day of the quarter on which class transition should occur; leave blank if unused.

QUARTERLY IN MONTH
Specify a number from 1 to 3 indicating the month in each quarter on which class transition should occur; leave blank if unused.

YEARLY ON DAY
Specify FIRST, LAST, or a number from 1 to 366 indicating the day of the year on which class transition should occur; leave blank if unused. For example, choosing the number 366 allows the transition to occur on 1/1 of the next year. In the event of a leap year, OSMC causes the transition to occur on 12/31 of the current year.

YEARLY IN MONTH
Specify a number from 1 to 12 indicating the month of the year on which class transition should occur; leave blank if unused.

Note: TIME SINCE CREATION, TIME SINCE LAST USE, and PERIODIC are mutually exclusive attributes and cannot be specified together.

An object's management class association can change as a result of an application request or a class transition. Should a change occur, OAM applies the new management criteria. This may result in a variety of actions, such as:
- Up to two backup copies may be made where one did not exist before.
- An object's lifetime may be decreased or increased.
- A new class transition event can cause the invocation of ACS routines in the future.

As you define management classes and prepare and review your implementation of class transition using the Automated Class Selection routines, it is critical to analyze the end result of your class transitions to avoid processing inefficiencies, unexpected results, or both.

The usage of TIME SINCE CREATION and TIME SINCE LAST USE attributes must be carefully studied to ensure that one of the class transitions in a series of class transitions assigns a management class, which causes the next class transition to occur **in the future** or the object to expire. Ensuring a management class is assigned to cause the next class transition to occur in the future is accomplished through your extensions to the operating system in the Management Class Automatic Class Selection routine.

Note: Do not use the TIME SINCE LAST USE or the TIME SINCE LAST REFERENCE attributes if you are using the new parameter (UPD=N) on the CBRINIT line in the IEFSSN*xx* PARMLIB member with no pending action date. See "Updating the IEFSSNxx PARMLIB member" on page 88 for more information.

If your implementation allows for a series of class transitions that do not result in a class transition scheduled in the future, or do not result in an object expiring and being deleted, the results of the storage management cycle may be affected. Depending on the number of objects processed, operational conditions, or possible processing interruptions due to contention, it is likely that processing will be seriously degraded. This could potentially force the storage management process into a loop that attempts to identify a future date for class transition processing or expiration for one or more objects.

If at any time an object's management class results in the object's expiration date being set to 9999/12/31 while that object is on removable media, that volume's expiration date will be set to 9999/12/31. This will cause the volume to never expire, even if the object's management class changes at a later date allowing the object to expire. Be aware of the affects of expiration dates that can be set by a management class, even if it is being used as an interim management class for an object. This expiration date can be modified using the MODIFY OAM,UPDATE command.

Defining Automatic Class Selection

41 ***Define and test ACS routines.***

You must perform this step at initial installation. During migration, you may optionally perform this step if you have made changes in the SMS definitions and programs used by OAM.

You must supply ACS routines. “Automatic Class Selection” on page 436 contains listings of source code for sample ACS routines. There can be only one set of ACS routines and exits in an active configuration. This set applies to both objects and data sets. Installation exits are optional and allow you to perform functions that are not permitted in the ACS routines (for example, writing GTF records).

Refer to *z/OS DFSMSdfp Storage Administration Reference* for information on using ISMF to define ACS routines. Consult *z/OS DFSMS: Implementing System-Managed Storage*; *MVS/ESA SML: Managing Storage Groups*; and *z/OS DFSMS Installation Exits* for information on writing ACS installation exits.

Validating and Activating the Configuration

42 ***Validate and activate the configuration.***

You must perform this step at initial installation. During migration, you may optionally perform this step if you have made changes in the SMS definitions and programs used by OAM.

Refer to *z/OS DFSMSdfp Storage Administration Reference* for information about validating and activating the configuration that you have just defined. You cannot use OAM until a configuration containing all the elements described in this chapter have been defined and activated.

Only one SCDS can be activated at any time. Activating another SCDS or reactivating the current SCDS while OAM is running may cause OAM to restart. How soon OAM is notified of the SCDS activation depends on the time interval specified with the INTERVAL keyword in the IGDSMS*xx* PARMLIB member. OAM restarts if RESTART=YES is specified on the OAM procedure JCL parameter. During this reinitialization, all libraries and drives are set to either online or offline according to the attributes defined in the SCDS that caused OAM to restart. After the restart completes, all libraries and drives should be displayed and set to the desired operational status.

Verifying Object Support Installation With IVP

43 ***Run the OAM Installation Verification Program for Object Support***

You must perform this step at initial installation and at migration.

The OAM Installation Verification Program (OAMIVP) is used to verify that OAM object support is successfully installed and operational. This program activates the OSREQ macro, and allows dummy objects to be tested by having the storage administrator perform OSREQ functions against them without having to perform an explicit OSREQ ACCESS to connect the macro to OAM, or without having to perform an OSREQ UNACCESS to disconnect the macro. Run this job (see “OAM Installation Verification Program and OAMUTIL” on page 433 for a sample of this job), and perform some OSREQ functions as a test to insure the product is successfully installed.

Moving OAM from One System to Another

It may be necessary to move the OAM application from one system to another to accommodate changes within an installation's storage management policy. The following information can assist you in moving OAM from one system to another (SYS1 and SYS2 are used as example system names).

Procedures for Transferring OAM to Another System

This section provides information on moving OAM from one OAM system to another OAM system, *neither of which is part of an OAMplex*. If you wish to merge OAMs into an OAMplex, you can run the CBRSMERG and the CBRSG100 jobs. CBRSMERG merges OAM configuration databases for use with DB2 data sharing. CBRSG100 performs a merge of the collection name catalogs and also the DB2 object databases in support of DB2 data sharing. For more information on these sample jobs, see "CBRSMERG" on page 415 and "CBRSG100" on page 422.

The following steps are necessary when moving the entire OAM application from one single system to another single system. These steps can be followed after the installation and migration steps have been completed, should you need to move OAM from one system to another.

__ Step 1. Install OAM and DB2 on system SYS2 by following the procedures in "Installation and Migration Checklist" on page 79.

__ Step 2. Make copies of the OAM DB2 databases from system SYS1 using DB2 utilities. The OAM DB2 databases are the Object Storage Databases, Object Administration Database, and the Optical Configuration Database described in "Appendix C. Understanding Databases for OAM Diagnosis" on page 453.

__ Step 3. Copy the Integrated Catalog Facility (ICF) catalog where the collections reside from system SYS1 using the Export function of Access Method Services (AMS). For more information on this Export function, see *z/OS DFSMS Access Method Services*.

__ Step 4. Start DB2 on system SYS2.

__ Step 5. Move the copies of the OAM DB2 databases to system SYS2 using DB2 utilities.

__ Step 6. Move the copies of the collection definitions to the Integrated Catalog Facility (ICF) on system SYS2 using the Import function of Access Method Services (AMS).

__ Step 7. Delete all of the 9246 optical library and drive definitions from system SYS1 using the ISMF Optical Library Application.

__ Step 8. If system SYS2 is not defined in the SCDS, then from system SYS1 add it as a valid system under the Control Data Set Definition Application of ISMF.

| __ Step 9. Move the SCDS from system SYS1 to system SYS2 by using the copy function of DFSMSdss. This only needs to be done if system SYS2 is not part of the same complex that is sharing the SMS SCDS.

__ Step 10. Using the ISMF Storage Group Application on system SYS2, alter the Object and Object Backup storage group definitions to change the connectivity of SYS1 to not connected and the connectivity of system SYS2 to enabled.

__ Step 11. Using the ISMF Optical Library Application on system SYS2, alter the library definitions connectivity from system SYS1 to system SYS2. This is only applicable for the controlling libraries. Altering these models automatically alters any connected library expansion units associated with the controlling libraries.

Note: You may need to delete and redefine all the 3995 operator-accessible drives.

__ Step 12. Using the ISMF Optical Library Application on system SYS2, redefine all of the 9246 library definitions. Using the ISMF Optical Drive Application on system SYS2, redefine all of the 9247 drive definitions associated with the 9246 libraries. Note that if the definitions do not exist in the DB2 Optical Configuration Database Tables that they will be added, but if they do exist, the information will just be updated in the DB2 Tables. The information will be added to the SCDS.

__ Step 13. Activate this SCDS from system SYS2. This may be done from the operator console by using the SETSMS SCDS (source control data set name) command or by using the Activate Configuration option of ISMF.

__ Step 14. Start OAM.

| Perform the following two steps if you wish to merge OAMs into an OAMplex.

| __ Step 1. Modify and run the CBRSMERG SAMPLIB job. See “Run the
| CBRSMERG SAMPLIB job” on page 121 for details.

| __ Step 2. Modify and run the CBRSG100 SAMPLIB job. See “Run the
| CBRSG100 SAMPLIB job” on page 122 for details.

Chapter 4. Administering OAM

This chapter discusses typical OAM administrative tasks, such as:
- Monitoring and maintaining the optical configuration using ISMF
- Monitoring and maintaining optical volumes
- Monitoring and maintaining SMS constructs and definitions
- Monitoring DB2 databases
- Tuning OAM
- Measuring OAM transaction performance using SMF
- Identifying transaction activity using RMF
- Establishing recovery procedures
- Using the Move Volume utility
- Processing expiration of objects
- Destroying and deleting expired data

Attention: Unless OAM is completely stopped, do not do any of the following:
- Stop or start table spaces or indexes related to the OAM databases
- Start DB2 in maintenance mode
- Run DB2 utilities against the OAM related databases
- Update any of the DB2 tables related to the OAM databases

Monitoring and Maintaining the Optical Configuration using ISMF

ISMF Library Management makes it possible to monitor and maintain information associated with the optical configuration and the source control data set. The following information concerning ISMF deals only with its role in an optical storage environment. ISMF is used in the management of tape libraries and their volumes; however, it is not discussed in conjunction with the information presented in this manual in an attempt to prevent redundancy of material found in the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ISMF Library Management

The library management dialog allows you to generate lists of libraries or drives, or to display the attributes of a single library or drive. It can be used to alter the definitions that were defined when OAM was installed or to add new definitions. For example:

Optical Library Configuration:
- Add, remove, or alter libraries and library drives
- Add, remove, or alter stand-alone or operator-accessible drives
- Redefine channel attachment

Typical ISMF Library Management Procedures

This section discusses the effects of some typical configuration maintenance tasks. Refer to *z/OS DFSMS: Using the Interactive Storage Management Facility* for detailed information about using ISMF.

Defining an Optical Library or Optical Disk Drive

The first time an optical library or optical disk drive is defined, the corresponding ISMF DEFINE panel requires that the storage administrator enter all the attributes for that device in the appropriate panel fields based on the name of the library or drive. When the storage administrator enters all the information, OAM adds a row containing that information to the library or drive table in the optical configuration database. An optical library or optical disk drive definition is also added to the specified SCDS in which the DEFINE occurs.

Subsequent definitions of the same optical library or optical disk drive into a new SCDS result in the REDEFINE panel being displayed with all the attributes associated with that device displayed as read in from the library or drive table. The result of this definition is that the optical library or optical disk drive definition will be added to the specified SCDS. For more detail on defining optical libraries and optical disk drives with ISMF, see “Sample ISMF Session for an IBM 3995 Optical Library Dataserver” on page 351.

SCDS Activation and Restart

Only one SCDS can be activated at any time. Activating another SCDS or reactivating the current SCDS while OAM is running may cause OAM to restart. During this reinitialization, all libraries and drives are set to either online or offline according to the attributes defined in the SCDS that caused OAM to restart. After the restart completes, all libraries and drives should be displayed and set to the desired operational status.

Deleting an Optical Library or Optical Disk Drive

Deleting an optical library or optical disk drive definition does not cause the entries to be deleted from the optical configuration database. Instead, upon deletion, the device definition will be removed from the specified SCDS. If you must delete an optical library or optical disk drive from the optical configuration database, use QMF, SPUFI, or a similar tool. For more detail on deleting optical libraries and optical disk drives with ISMF, see “Deleting an Optical Library” on page 377, and “Deleting an Optical Disk Drive” on page 378.

Altering an Optical Library or Optical Disk Drive

Altering an optical library or optical disk drive results in an update to the corresponding database row to reflect the changes. Within the specified SCDS, the online status of the definition associated with the device is updated to reflect any change made to the online status. For more detail on altering optical libraries and optical disk drives with ISMF, see “Altering a 3995 Optical Library” on page 368, and “Altering an Optical Disk Drive” on page 375.

Note: The function used to alter an optical library or optical disk drive definition is only available when OAM is *not* running. This prevents OAM from overwriting a change initiated from an ISMF Library Management dialog.

Auditing an Optical Library

You can use the AUDIT line operator on the Optical Library List to perform inventory tasks against an entire 3995 optical library. The physical location of all the optical volumes associated with a 3995 optical library (full library audit) can be verified using the AUDIT line operator. The 9246, 3995-C3A, and pseudo optical libraries cannot be audited.

| **Note:** You can now perform the AUDIT function using a new operator command.
| See “Auditing a Volume” on page 317 for more information.

For information concerning auditing a tape library using the AUDIT line operator, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Because the library audit may take a long time, a confirmation panel is displayed, asking you to confirm the AUDIT request. To confirm, type in “Y”, then press ENTER. See Figure 19 on page 153 for the Confirm Audit Request panel.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                           CONFIRM AUDIT REQUEST
Command ===>

To Confirm Audit of the Following Library:

   Library Name:  LIB1


   Enter "/" to select option      Perform Audit



   Note: If Audit is performed, Audit requests will be interspersed with other
         requests to the library. Audit may take a long time to complete.
         Use the HELP command for more information on Audit processing.





Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 19. Confirm Library Audit Request Panel

AUDIT execution can be a lengthy process. During AUDIT execution, other activity in the library is **not** quiesced, and OAM prioritizes AUDIT requests lower than other requested functions. The following activities are examples of activities that OAM can still process while AUDIT is in progress:

- Requests to read objects from optical volumes within the library that is being audited.
- Requests to write objects to optical volumes within the library that is being audited.
- Optical volume ejects from the library that is being audited.

It may take **several** hours for you to receive notification that a full library audit or an extensive volume list audit has completed. Therefore, when scheduling an audit, be sure to take work load and time factors into consideration.

When you receive a completion message, you can enter the LISTVOL line operator to display the Mountable Optical Volume List for all optical volumes in the audited library. Check the VOLUME ERROR STATUS column for the results of the audit. OAM also issues messages for errors found. If you log off the Time Sharing Option (TSO) session before completion, the messages are stored in a broadcast data set and displayed the next time you log on. See Table 26 on page 163 for information on the AUDIT completion messages and the results displayed in the VOLUME ERROR STATUS column.

Remapping an Optical Library

If audits or other functions indicate volume locations are incorrect, you can use the REMAP line operator to reconstruct the optical library’s outboard inventory, by verifying the identity of each volume in the optical library. (The outboard inventory is maintained by the controller in the IBM 3995 Optical Library Dataserver.) REMAP also detects and corrects any discrepancies between the outboard inventory and the volume table in the optical configuration database. The REMAP line operator is not supported in conjunction with tape libraries.

An operator command performs the REMAP function. See “Remapping an Optical Library” on page 318 for more information on the syntax and parameters for this command.

Enter the REMAP line operator next to a 3995 optical library on the Optical Library List. REMAP checks for the following conditions:
- The CDS name is “ACTIVE”.
- The optical library is a real 3995 library. (You cannot use REMAP with a 9246 library, a 3995-C3A or a pseudo library of any kind.)
- The optical library is online and operational.
- At least one internal drive is online and operational. The drive cannot be pending offline.

If these checks are successful, the REMAP Confirmation panel is displayed, asking you to confirm the REMAP. To confirm, type in “Y”, then press ENTER. If the REMAP is accepted, the message REMAP SCHEDULED is displayed. Once the REMAP process is started, it must run to completion. There is no REMAP cancel function.

Note: REMAP execution can take from 30 to 80 minutes to complete a full library depending on the number of online drives in the library. Therefore, when scheduling a REMAP, be sure to take work load and time factors into consideration. Also, once a REMAP is started, stopping OAM WILL NOT cancel REMAP processing. If you stop OAM in an attempt to cancel or stop a REMAP request, the library controller inventory must be rebuilt, and the REMAP continues to process in the library. There is no option of canceling a REMAP request once it is started.

When a REMAP request for a library is issued, all work in progress for the library is allowed to complete; however, new work requests and work requests scheduled but not yet started are rejected (with the exception of drive vary requests that were queued prior to the REMAP request). Cartridges in the internal drives are demounted. If the library has an operator-accessible drive, upon completion of work in progress, the cartridge is demounted and the drive becomes unavailable for activity until the REMAP processing is complete. If an internal drive cartridge demount fails, REMAP processing fails; however, if the demount fails on the operator-accessible drive, REMAP processing continues on the internal drives.

Once REMAP is issued, the library controller reconstructs its inventory by going to each slot in the library. It then verifies the identity of the cartridge in that slot by inserting that cartridge in a drive and reading both volume serial numbers. When the new inventory is complete, the host reads the inventory and compares all the inventory records with the volume records, and the volume records with the inventory. Updates are made to the appropriate host tables or the cartridge is ejected if the table cannot be corrected. Host volume records that indicate a volume is in the library but cannot be found in the outboard inventory are marked as shelf-resident and the volume status is updated as lost. All duplicate cartridges are marked by the library controller, causing the host to schedule the cartridge for eject. You are notified of the start of REMAP, all ejected cartridges, and of the completion of REMAP through a message to your TSO logon session.

When you receive a completion message, you can enter the LISTVOL line operator to display the Mountable Optical Volume List for all optical volumes in the remapped library. Check the VOLUME ERROR STATUS column, described in Table 24 on page 155, for error messages issued for errors detected by the REMAP operation.

Table 24. Remap Results That Appear in the VOLUME ERROR STATUS Column

Result	Meaning
NO ERROR	Either no error occurs during the REMAP, or shows the initial status of the VOLUME ERROR STATUS column.
LOST VOLUME	A record in the volume table indicates a volume is in the optical library, but the volume cannot be found in the outboard inventory for that optical library.

The error messages are issued to the ISMF user who invoked the REMAP line operator. The messages contain the serial number (volser) of the volume for which the error was found and text indicating the type of error found. If you log off the TSO session before completion, the messages are stored in a broadcast data set and displayed the next time you log on. REMAP also may update the VOLUME LOCATION, VOLUME LIBRARY NAME, and MEDIA TYPE columns.

Types of Discrepancies that REMAP Resolves

Table 25 shows how REMAP resolves various discrepancies.

Table 25. Discrepancies REMAP Resolves between the Outboard Inventory and the Optical Configuration Database

Cause	Resolution
A cartridge has a media error. Possible causes are damaged media, the volume is a duplicate, or the cartridge is unformatted.	The cartridge is ejected from the optical library to allow it to be inspected.
The row in the volume table in the OCDB indicates the volume should be in another real optical library.	The volume is ejected from this library.
The paired volumes of a cartridge do not match the paired volumes in the OCDB.	The cartridge is ejected.
A volume physically resides in the library, but no row for that volume is found in the OCDB.	The cartridge is ejected.
The row in the volume table in the OCDB indicates the volume is on the shelf, but is physically residing in the library.	The volume location is changed to state the volume is residing in the library. The library name also is corrected if it differs from the library in which the volume resides.
The OCDB has a row for the volume but the outboard inventory does not have the corresponding volume.	The volume is lost. The VOLUME ERROR STATUS column for the volume is updated to say “LOST VOLUME”. The volume is given a pseudo library name. The volume location is changed to “SHELF”.
The volume media type in the OCDB is not compatible with the library device type.	The media type is corrected.

Monitoring and Maintaining Optical Volumes

The ISMF Volume function assists in the maintenance and verification of the optical volumes within the optical library through the use of the Mountable Optical Volume Application available from the Volume List Selection menu.

For information concerning ISMF and the Mountable Tape Volume Application used in conjunction with tape libraries and volumes, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

ISMF Mountable Optical Volume Application

A mountable optical volume resides on one side of an optical disk cartridge. The ISMF Mountable Optical Volume Application allows you to maintain optical volumes in an optical library through the use of the AUDIT, EJECT, and RECOVER line operators, or through the use of the AUDIT command. You can use line operators and commands to perform audits against volume lists, single volumes, or full libraries, eject optical disks from the library, and recover data stored on an optical disk that can no longer be read.

Note: The ISMF RECOVER line operator does not support recovery from second backup copies of objects. In order to exploit recovery of a primary volume from the second object backup copy, the MODIFY OAM command must be used. See "Starting Recovery Functions" on page 229 for further details on this procedure.

Generating a Mountable Optical Volume List

With the Mountable Optical Volume Application, you can generate a volume list of optical volumes for the following types of optical media:

9247 single-density, WORM
3995 5.25-inch, single-density, WORM
3995 5.25-inch, double-density, WORM
3995 5.25-inch, quad-density, WORM
3995 5.25-inch, 8x-density, WORM
3995 5.25-inch, single-density, rewritable
3995 5.25-inch, double-density, rewritable
3995 5.25-inch, quad-density, rewritable
3995 5.25-inch, 8x-density, rewritable

Note: Double-, quad-, and 8x-density WORM also includes CCW media. CCW is continuous composite WORM media. WORM is write-once-read-many media.

Completing the Mountable Optical Volume Selection Entry Panel

Follow these steps to bring up the Mountable Optical Volume Selection Entry panel.

1. Select option 2, VOLUME, from the ISMF Primary Option menu (see Figure 36 on page 328). ISMF displays the Volume List Selection menu.
2. Select option 2, MOUNTABLE OPTICAL, to generate a list of mountable optical volumes. ISMF displays the Mountable Optical Volume Selection Entry panel, shown in Figure 20 on page 157.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                MOUNTABLE OPTICAL VOLUME SELECTION ENTRY PANEL
Command ===>

Select Source to Generate Volume List . . 2   (1 - Saved list, 2 - New list)

     1 Generate from a Saved List
         List Name  . .

     2 Generate a New List from Criteria Below
         Volume Serial Number . . . *         (fully or partially specified)
         Library Name . . . . . . . *         (fully or partially specified)
         Storage Group Name . . . . *         (fully or partially specified)
         Optical Media Type . . . . ALL       (See help for valid value)



       Enter "/" to select option     _ Respecify View Criteria
                                      _ Respecify Sort Criteria

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 20. Mountable Optical Volume Selection Entry Panel

This section describes only the fields used to generate a new list (option 2). *z/OS DFSMS: Using the Interactive Storage Management Facility* describes all the columns in the Mountable Optical Volume Selection Entry panel.

GENERATE A NEW LIST FROM CRITERIA BELOW

Complete the following fields if you selected option 2 (default) for SELECT SOURCE TO GENERATE VOLUME LIST:

VOLUME SERIAL NUMBER

Enter a full or partial serial number of the volume or volumes to include in the list. The default value is an asterisk.

To include a single volume, enter a fully qualified volume serial number of 1 to 6 characters:

```
       VOLUME SERIAL NUMBER ===> SYS001
```

For a partially qualified volume serial number, use asterisks as global volume serial number characters or percent signs as place holders.

For example, to include a range of volumes, enter a partially qualified volume serial number by using one or two asterisks as global volume serial number characters:

```
      VOLUME SERIAL NUMBER ===> SYS*
```

Two asterisks are the maximum number of volume serial number characters allowed.

| Use a single asterisk to specify all volumes that fit your other selection
| criteria:

```
VOLUME SERIAL NUMBER ===> *
```

This field is primed with the last value used. The default is *

LIBRARY NAME
Enter the 1 to 8 character name of an optical library, or a partially qualified name. This field is primed with the last value used. The default value is an asterisk.

STORAGE GROUP NAME
Enter the 1 to 8 character name of an SMS Object or Object Backup storage group, in the same way as you would for a volume serial number. This field is primed with the last value used. The default value is an asterisk.

OPTICAL MEDIA TYPE
Enter the 3 to 8 character name of the optical media type on which the volume resides. **ALL** is the default value. This field is primed with the last value used. Use the following values for optical media types:

9247	9247 optical disk volumes
3995WORM	3995 single-, double-, quad-, and 8x-density, WORM, optical disk volumes
3995REWR	3995 single-, double-, quad-, and 8x-density, rewritable, optical disk volumes
3995-1	3995 single-density, WORM and rewritable, optical disk volumes
3995-1WO	3995 single-density, WORM, optical disk volumes
3995-1RW	3995 single-density, rewritable, optical disk volumes
3995-2	3995 double-density, WORM and rewritable, optical disk volumes
3995-2WO	3995 double-density, WORM, optical disk volumes
3995-2RW	3995 double-density, rewritable, optical disk volumes
3995-4	All 3995 quad-density, rewritable or WORM, optical disk volumes
3995-4RW	All 3995 quad-density, rewritable, optical disk volumes
3995-4WO	All 3995 quad-density, WORM, optical disk volumes
3995-8	All 8x-density, rewritable or WORM, optical disk volumes
3995-8RW	All 3995 8x-density, rewritable, optical disk volumes
3995-8WO	All 8x-density, WORM, optical disk volumes
ALL	Select all available optical media types.

Note: Double-, quad-, and 8x-density WORM includes CCW media. CCW is continuous composite WORM media. WORM is write-once-read-many media. REWR is rewritable media.

Final Step: Generating the List

After entering the information you want on the Mountable Optical Volume Selection Entry panel, you are ready to generate the list. Press the ENTER key to display the volumes that meet your selection criteria. Figure 21 on page 159 and Figure 22 on page 159 show columns 14 through 21 of the Mountable Optical Volume list. Scroll the list to see all the columns.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                        MOUNTABLE OPTICAL VOLUME LIST
Command ===>                                                  SCROLL ===> HALF
                                                       Entries 1-7 of 7
Enter line operators below:                            Data Columns 14-16 of 21
    LINE       VOLUME                              MEDIA
    OPERATOR   SERIAL  SHELF LOCATION              TYPE      VOLUME ERROR STATUS
   ---(1)----  -(2)--  -----(14)---------------  --(15)--  ---------(16)-------
               SYS090  COMPUTER CENTER LIB 7     3995-1WO  NO ERROR------------
               SYS092  COMPUTER CENTER LIB 7     3995-1RW  NO ERROR------------
               SYS093  ------------------------  3995-1RW  --------------------
               SYS095  OFFSITE                   3995-2RW  VOLUME NOT FOUND----
               SYS096  SHELF 2                   3995-2WO  WRONG VOLUME IN SLOT
               SYS097  SHELF 2                   3995-2WO  NO ERROR------------
   ----------  ------  -----------  BOTTOM OF DATA  -----  --------------------




Use HELP Command for Help; Use END Command to Exit.
```

Figure 21. Mountable Optical Volume List, Columns 14 through 16

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                        MOUNTABLE OPTICAL VOLUME LIST
Command ===>                                                  SCROLL ===> HALF
                                                       Entries 1-7 of 7
Enter line operators below:                            Data Columns 17-21 of 21

    LINE       VOLUME  CAPACITY  VOLUME       ENTER OR    PSEUDO    OAM INSTANCE
    OPERATOR   SERIAL  (IN MB)   CREATE DATE  EJECT DATE  LIB NAME  MEMBER NAME
   ---(1)----  -(2)--  --(17)--  ---(18)---  ---(19)---   --(20)--  ---(21)-----
               SYS090       320  1992/07/29  1992/07/29   --------  ------------
               SYS092       320  1992/10/22  1992/10/22   --------  ------------
               SYS093       320  1992/10/22  1992/11/22   --------  ------------
               SYS095       640  1992/11/04  1992/11/04   --------  ------------
               SYS096       640  1992/11/10  1992/11/10   PLIB2     OAMIMEMBER1
               SYS097       640  1992/12/20  1992/12/20   PLIB1     OAMIMEMBER2
   ----------  ------   -------  BOTTOM  OF  DATA  ----   --------  ------------



Use HELP Command for Help; Use END Command to Exit.
```

Figure 22. Mountable Optical Volume List, Columns 17 through 21

Refer to *z/OS DFSMS: Using the Interactive Storage Management Facility* for information on columns 1–13 on the Mountable Optical Volume list.

SHELF LOCATION
The physical location of the optical volume that resides outside (shelf-resident) of an optical library.

MEDIA TYPE
Displays the type of optical media upon which an optical volume resides. The MEDIA TYPE field applies to all optical volumes. The valid values are:

9247 9247 optical disk media
3995-1WO 3995 single-density, WORM, optical disk media
3995-1RW 3995 single-density, rewritable, optical disk media
3995-2WO 3995 double-density, WORM, optical disk media

3995-2RW 3995 double-density, rewritable, optical disk media

3995-4RW 3995 quad-density, rewritable, optical disk media

3995-4WO 3995 quad-density, WORM, optical disk media

3995-8WO 3995 8x-density, WORM, optical disk media

3995-8RW 3995 8x-density, rewritable, optical disk media.

UNKNOWN The REMAP function encountered a MEDIA TYPE of “9247” when it processed volumes in a 3995 library. “UNKNOWN” only occurs in conjunction with the display of “LOST VOLUME” in the VOLUME ERROR STATUS column.

???????? If the value cannot be displayed because of an error, the following columns (which depend upon a valid MEDIA TYPE) display question marks: FREE SPACE, %USED, and VOLUME ERROR STATUS

Note: Double-, quad-, and 8x-density WORM includes CCW media. CCW is continuous composite WORM media. WORM is write-once-read-many media.

VOLUME ERROR STATUS
Shows the error status of individual optical volumes. The VOLUME ERROR STATUS shows the status after the volume is audited, or the volume status after the remapping of a 3995 optical library.

CAPACITY (IN MB)
Shows the raw unformatted capacity in megabytes (1 MB = 1 048 576 bytes) of the optical disk volume.

VOLUME CREATE DATE
Shows the date that the volume was created and initially labeled in the form YYYY/MM/DD, where YYYY is the year, MM is the month of the year, and DD is the day of the month.

ENTER OR EJECT DATE
The date that the volume was entered into the optical library if the volume is currently library-resident. If the volume is currently shelf-resident, this is the date that the volume was last ejected from an optical library. The format is YYYY/MM/DD, where YYYY is the year, MM is the month of the year, and DD is the day of the month.

PSEUDO LIB NAME
The name of the pseudo library if the volume is a shelf-resident volume. This field should match the library name field when the volume is shelf-resident. If the volume is library-resident, this is the target pseudo library to which the volume is assigned when it is ejected from the real library.

OAM MEMBER INSTANCE NAME
The instance of OAM in an OAMplex that is currently managing and controlling this volume. If the volume is library-resident, this is the member name associated with the OAM where the optical library is currently physically online. If the volume is shelf-resident and the member name is not blank, the volume is currently mounted on an operator-accessible drive, which is currently online to the OAM identified by this member name.

Viewing and Sorting a List

You can sort and tailor a list with the View and Sort options on the Mountable Optical Volume List Selection Entry panel. *z/OS DFSMS: Using the Interactive Storage Management Facility* discusses the View and Sort options in more detail.

Maintaining and Verifying the Volume List

You can use the AUDIT, EJECT, and RECOVER line operators, and the AUDIT command, to maintain and verify optical volumes in your optical library, and if errors are found, you can reconstruct an accurate list using the REMAP line operator (see “Remapping an Optical Library” on page 153 for more information on this line operator). The line operators affect individual volumes. In contrast, the commands affect all eligible volumes on the Mountable Optical Volume list, except for the optical volumes that you choose to hide. You can issue an audit from the host application through ISMF Library Management (see “Auditing an Optical Library” on page 152 for more information) or Mountable Optical Volume Application.

Note: The ISMF RECOVER line operator does not support recovery from second backup copies of objects. Use the MODIFY OAM command to exploit recovery of a primary volume from the second object backup copy. See “Starting Recovery Functions” on page 229 for further details on this procedure.

Press PF1 on the Mountable Optical Volume list for help information about the line operators and list commands.

Verifying Optical Volumes Using Audit

AUDIT can be invoked as an ISMF line operator on the Mountable Optical Volume List panel (Single Volume AUDIT) or the Optical Library List panel (Full Library AUDIT). The storage administrator uses the AUDIT line operator to verify the physical location of an optical volume. The audit compares the volume information maintained in the optical configuration database and the optical library outboard inventory with the actual location of the optical volume. AUDIT does not just compare the optical configuration database with the outboard inventory; it actually causes the volume to be mounted and reads the volume serial number to verify that the volume is in its assigned storage location.

Note: The AUDIT function can now be performed via a new operator command. See “Auditing a Volume” on page 317 for more information.

For information concerning using the AUDIT line operator with tape libraries, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

When you invoke the AUDIT line operator successfully, AUDIT SCHEDULED is displayed on the Mountable Optical Volume list. If the volume is successfully scheduled for an audit, the volume has ***AUDIT** displayed in the line operator column. Audits that are not successfully scheduled have **¬AUDIT** or **?AUDIT** in the line operator column. ISMF also displays a short message explaining why the audit is not scheduled (see Table 28 on page 164 for more information).

AUDIT can also be invoked as an ISMF command to audit all the eligible optical volumes on the Mountable Optical Volume List (Volume List AUDIT). ISMF is an important part of the AUDIT scheme because it allows you to start with an entire optical volume list, and by using sorting and filtering capabilities, you reduce that list to a subset of volumes; for example, all the volumes in storage group *x*. You can then use the AUDIT command to request an audit of all volumes in that subset list. (The AUDIT command does not affect volumes that you have hidden using the HIDE line operator.)

AUDIT functions help you ascertain the physical location of optical volumes by verifying whether or not a library volume resides in the location that is listed for that volume in the optical library outboard inventory. The library controller maintains the internal library location of the cartridges in the outboard inventory. The host also identifies which library contains each cartridge in the volume table. If the host record or records do not match the controller inventory when OAM performs an audit, then you must correct the records, the inventory, or both. The AUDIT functions do not perform any corrective actions; their purpose is verification only. See “Remapping an Optical Library” on page 153 for a description of corrective actions that you can take if the audit is unsuccessful.

Notes:

1. The AUDIT function is available only for 3995 (except for 3995-C3A) libraries and is not used in conjunction with the 9246 or pseudo optical libraries.
2. AUDIT functions only process volumes that are known to OAM. If an unknown volume is in the library, the AUDIT functions do not detect this condition. REMAP functions are necessary to locate and eject the unknown volume.
3. AUDIT functions from the host application and the 3995 dynamic console are not the same. In an Multiple Virtual Storage (MVS) installation, the AUDIT function that is available from the 3995 dynamic console is intended for use during hardware service; AUDIT functions that are issued from the 3995 dynamic console do not communicate with OAM at the host. The AUDIT function from the 3995 dynamic console performs the same hardware function as a library REMAP. In an MVS installation, issue the AUDIT functions from the host application and not from the 3995 dynamic console; otherwise, the host and the library controller can differ. This manual describes only the command functions available from the host application through ISMF Library and Volume applications and MVS operator commands. You can find further information about commands from the 3995 dynamic console in the *3995 Operator’s Guide* or the *3995 Operator’s Guide C-Series Models*.

ISMF AUDIT provides three scopes:
- Single volume audit (AUDIT line operator)
- Volume list audit (AUDIT command)
- Full library audit (AUDIT line operator)

For each volume audited, three conditions must be present:
- Volume must be obtained from its assigned storage slot.
- Volume must be mounted.
- Internal label of the volume must be read and verified as matching the entry for that storage slot/volume combination in the outboard inventory.

When you receive an audit completion message, you can use the REFRESH command to update the Mountable Optical Volume list with the same selection criteria. The results of the audit are shown in the VOLUME ERROR STATUS column and you also receive a message with the error results. If you log off before the results are obtained, messages are stored in the users broadcast data set and displayed during the next logon process. VOLUME ERROR STATUS contains only the last error found; no history is kept. No attempts are made to fix the problems at the time of detection because, based on the error found, the software is unable to determine exactly what the corrective action should be. Use REMAP, discussed in “Remapping an Optical Library” on page 153, to correct the problems. Table 26 on page 163 lists possible results of auditing an optical volume.

Table 26. Auditing Results That Appear in the VOLUME ERROR STATUS Column

Result	Meaning
NO ERROR	Either no error occurs during the audit, or shows the initial status of the VOLUME ERROR STATUS column.
NOT IN THE LIBRARY	The volume has an entry in the optical configuration database, but no corresponding entry in the outboard inventory.
VOLUME NOT FOUND	The volume was not found in its assigned storage slot. Either the slot is empty or another volume was found.
VOLUME IN WRONG SLOT	This volume was found while auditing another volume or when attempting to mount another volume.
MEDIA ERROR	An error occurred when the volume serial number was read.

Notes:
1. You cannot audit optical volumes that are on a shelf.
2. You cannot audit optical volumes in a 9246 Optical Library.

When you request an AUDIT, you must specify all volsers to be audited, even if two of the volsers represent opposite sides of the same cartridge. When you specify a full library AUDIT using the AUDIT line operator, all volume serial numbers listed in the host inventory as residing in that library are audited.

Whenever a full library AUDIT or a volume list AUDIT is requested, a confirmation panel is displayed. This panel gives you the opportunity to confirm or deny the audit request. To confirm, type in **Y**, then press ENTER. See Figure 23 for the Confirm Audit Request panel.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                           CONFIRM AUDIT REQUEST
Command ===>

Number of Volumes to be Audited: 97


Specify the Following:
   Enter "/" to select option      Perform Audit



   Note: If Audit is Performed, Audit Requests will be Interspersed with other
          Requests, with the Audit Request having low priority.







Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 23. Confirm Optical Volume Audit Panel

When you receive an audit completion message, you can use the REFRESH command to update the Mountable Optical Volume List with the same selection criteria. The results of the audit are shown in the VOLUME ERROR STATUS column, discussed in Table 26 on page 163. If you log off before completion, resulting messages are stored in your broadcast data set and displayed at the next logon.

AUDIT Messages

After confirming the audit request, a generic message is displayed on the Mountable Optical Volume List indicating whether all, some, or none of the optical volumes were scheduled for an audit. Table 27 explains the meaning of the generic AUDIT messages.

Table 27. Generic AUDIT Messages

Generic AUDIT Message	Meaning
AUDIT SCHEDULED	All optical volumes were successfully scheduled for audit.
SELECTED AUDIT SCHEDULED	Only the eligible optical volumes were scheduled by ISMF for audit. Optical volumes on the shelf or in a 9246 Optical Library are ineligible for audit.
PARTIAL AUDIT SCHEDULED	At least one optical volume was rejected by OAM and at least one optical volume was successfully scheduled for the audit.
NO VOLUMES SELECTED	None of the optical volumes were eligible for the audit by ISMF.
AUDIT FAILED	All the optical volumes were rejected for the audit by OAM.

The AUDIT command shows in the LINE OPERATOR column for each individual optical volume whether the audit was successfully scheduled, as shown in Table 28.

Table 28. Specific AUDIT Messages

AUDIT Message	Meaning
*AUDIT	The optical volume was scheduled for an audit.
¬AUDIT	The optical volume was not eligible for audit.
?AUDIT	The audit request for the optical volume was rejected.

The AUDIT command functions provide a message history for each optical volume, and ISMF log entries about rejected audit requests. See “Handling OAM Scheduling Errors” on page 165 for details.

Ejecting Optical Disks

The EJECT line operator schedules the mountable optical disk to be moved to the input/output station of the optical library. When you submit the line operator successfully, EJECT SUBMITTED is displayed on the Mountable Optical Volume List. After the eject completes, OAM issues a completion message.

When an optical disk is ejected, the operator may be prompted on the system console to supply a shelf location and a pseudo library to assign the ejected volume to. Ejected optical disks are stored according to the optical shelf conventions established at your computer site. The information on the shelf location and the pseudo library is stored in the OCDB for later use when the optical disk needs to be obtained from the shelf and a drive selected for its use. When the optical disk is obtained from the shelf, it is mounted within an operator-accessible or stand-alone drive, or within a library if no outstanding mount exists for the cartridge.

After receiving the EJECT completion message, you can use the REFRESH command to update and view the shelf location and pseudo library information for an optical volume in the SHELF LOCATION column. See “Ejecting an Optical Disk” on page 253 for more information on the EJECT command, including command syntax.

Recovering Optical Disks

The RECOVER line operator allows you to invoke the Volume Recovery utility to recover data stored on an optical disk that can no longer be read. When you successfully enter the RECOVER line operator against an optical volume serial number, RECOVER SUBMITTED is displayed on the Mountable Optical Volume list. Subsequent processing is the same as that performed when an operator command is used to invoke the Volume Recovery utility.

Before you enter the RECOVER line operator, become familiar with the prerequisites and the dialogue that occurs with the system operator. Refer to “Recovering an Entire Optical Cartridge or Tape Volume” on page 192 for more information on the Volume Recovery utility.

Note: The ISMF RECOVER line operator does not support recovery from second backup copies of objects. Use the MODIFY OAM command to exploit recovery of a primary volume from the second object backup copy. See “Starting Recovery Functions” on page 229 for further details on this procedure.

Handling OAM Scheduling Errors

The following information provides assistance in handling OAM scheduling errors.

Message History for AUDIT Commands

The AUDIT command or line operator provides a message history for each optical volume. Enter the MESSAGE line operator next to any optical volume serial number to display the short message for the specific volume. Press PF1 to display the additional long message.

ISMF Log Entries about Rejected Requests

OAM schedules AUDIT, REMAP, and EJECT requests. If OAM rejects the requests, or an error occurs during the scheduling of the request, the OAM errors are recorded as ISMF log entries. An example of a rejected request is an ?AUDIT in the LINE OPERATOR column for a specific optical volume. See the feedback area of the ISMF log entry for the return codes and reason codes for OAM errors. The ISMF log entries are in the ISPF Transaction Log. Refer to *z/OS DFSMSdfp Diagnosis Reference* for the meaning of the OAM return and reason codes.

Errors After a Request is Scheduled

If discrepancies are found after issuing an AUDIT command, they are noted and related to you by the following means:

- Scheduling error messages for full library audits are issued to you via the MVS SEND interface unless the scheduling error occurred prior to any volumes from the library being successfully scheduled and that error was severe enough to prevent any other volumes in the library from also being scheduled. This early detected severe error is reported on the ISMF panel via the help facilities offered by ISMF. SEND messages contain the volser (if known) of the volume for which the error was found and text that indicates the type of error found in attempting to validate an audit request.

  If the scope of the audit is volume list or single volume, scheduling errors are not reported to you via the MVS SEND interface. These errors are indicated on

return to the ISMF panel from which the AUDIT was initiated and can be interrogated by using the message and help facilities offered through the ISMF panels. Refer to *z/OS DFSMSdfp Storage Administration Reference* manual for more information about the ISMF message and help facilities.
- Errors incurred while attempting to perform the physical audit for any of the three audit scopes, single volume, volume list, or full library are reported to you via the MVS SEND interface.
- After auditing a volume, the error status field (ERRSTAT) of the optical configuration database (OCDB) volume record is updated. As notification that the audit is complete and the error status fields can be reviewed, a completion message is sent to you through your TSO logon session. If you are not logged on when OAM issues these errors or completion messages, they are saved in the SYS1.BRODCAST data set. You will receive these messages the next time you log on to TSO.

Monitoring and Maintaining SMS Construct Definitions

ISMF Library Management makes it possible to monitor and make changes to the SMS constructs. Refer to *z/OS DFSMS: Using the Interactive Storage Management Facility* for detailed information about using ISMF.

Changing SMS Construct Definitions

As installation requirements change, it may be necessary to update definitions of the storage classes, management classes, storage groups, data class, and ACS routines. Definitions for these constructs can be modified using the ISMF ALTER panels in an optical storage environment.

ISMF Alter panels are not used in support of object tape processing. SMS constructs for objects that reside on tape may be changed through the use of OSREQ STORE or CHANGE macro, as a result of a class transition, or by an ACS routine overriding information on the SETOAM statement. Although the ISMF function is not applicable, the concepts concerning monitoring and maintaining SMS constructs (for example, performance requirements, retention periods, class transitions, expiration dates, and such) do apply to objects stored on tape volumes, and should be considered in the following discussions.

These updates must be done with caution because objects that do not require processing after the definition is changed will not be affected by the change, even though they are assigned to the class to which the new definition applies. The updated definitions will be used *only* for objects entering the system, or processed by the system, after the change. This is particularly true in changes to management class definitions that affect retention, backup, or class transition.

Maintaining Storage Class Definitions

A new storage class may be needed to define the performance requirements of a new application. In addition to the definition of the new storage class, it will be necessary to add or modify management class definitions and the ACS routines to manage objects that use the new storage class.

Storage class definitions may be changed to accomplish the following functions:
- Add storage classes
- Alter the INITIAL ACCESS RESPONSE SECONDS to cause data to be stored on removable media (optical or tape volumes depending on the Sustained Data Rate) instead of DASD space.

Maintaining Management Class Definitions

When the need for a new object cycle is recognized, it usually leads to the definition of a new set of management classes for the phases in the new cycle. It becomes necessary to add statements to the ACS routine to process the transitions for the new management classes.

The effect of the changed management class definitions on objects that currently exist must be considered. The change of a management class may imply conversion action which is not supported by OSMC. For example, a transition rule could be changed to cause the schedule date to occur a month earlier. OSMC would not process the object until the scheduled date assigned using the previous transition rule, which is a month later than is specified by the updated definition.

Management class definitions may be changed to accomplish the following functions:
- Add management classes
- Alter backup requirements
- Change retention criteria
- Modify class transition parameters

Maintaining Object Storage Group Definitions

New Object storage groups may be needed for physical separation of new types of objects. In addition to defining the new Object storage groups, it may be necessary to change the installation's object naming conventions and to modify the ACS routines to use the new naming conventions to assign objects to the new storage groups.

Object storage group construct definitions may be changed to accomplish the following functions:
- Define a different storage management cycle start window
- Change DRIVE STARTUP threshold for optical
- Change the criteria used to determine when volumes are marked full
- Add an optical library
- Use new operator-accessible drives
- Remove an optical library (for migration to a newer library or media)
- Change the OSMC processing system name
- Change system connectivity

Object backup storage group construct definitions may be changed to accomplish the following functions:
- Change the target optical devices for backup data
- Add an optical library
- Change the criteria used to determine when volumes are marked full
- Remove an optical library
- Change DRIVE STARTUP threshold for optical
- Change system connectivity

Modifying Default Storage and Management Classes

Although IBM strongly advises against direct modification of the DB2 tables, it is sometimes necessary; therefore, it is recommended that relevant data be backed up before proceeding. Also, quiesce any system activity that might be active for the collection and/or storage group so the application does not receive errors if attempting to access the collection data while changes are made. The following procedure can be used to change the default storage class, management class, or both for an existing object storage group's collection ID.

1. Delete the collection definition from the catalog:

```
DELETE  COLLECTION_NAME NONVSAM CATALOG ('CATALOG_NAME') –
        FILE (DD1)  PURGE NOSCRATCH
```

2. Change the storage class, or management class, or both in the DB2 collection table to the desired value.

   Note: It is the installation’s responsibility to ensure that these are valid values in the SMS CDS.
3. Add the collection entry back to the catalog:

   ```
   DEFINE NVSAM (NAME(COLLECTION_NAME) COLLECTION RECATALOG)
   ```

   This creates a collection entry, but it will not contain a management or storage class.
4. Use OSREQ RETRIEVE to retrieve an existing object in the collection, or OSREQ STORE to store a new object into the collection. This causes an update of the catalog version. See *z/OS DFSMS OAM Application Programmer’s Reference* for details on using the OSREQ macro.
5. Verify that the catalog matches DB2.

   ```
   LISTCAT CATALOG('CATALOG_NAME') FILE(DD1) ALL NVSAM
   ```

   Note: The new default storage class, or management class, or both only applies to new objects stored after these changes are complete.

Changing ACS Routines

As mentioned above, ACS routines may need to be changed to support changes in storage group, management class, storage class, or data class definitions. Defining new storage groups, storage classes, management classes, and data classes has no effect unless the ACS routines are changed to select those new constructs.

ACS routines can be changed to accomplish the following functions:
- Provide initial class defaults for new collections
- Cause an object to move differently in the hierarchy by assigning a different storage class at class transition

Note: Defining new classes does not always mean new values for parameters; a new class can have the same parameters as an existing class. A new class may be created to make the relationship between a class and an application more understandable. This action makes it possible to modify parameters later to fit the needs of one application without affecting other applications. For example, adding a new management class that has the same backup parameter as an existing class allows you to change the backup parameter later for the new application’s objects without changing the backup requirements for other objects associated with the original class.

A Final Note of Caution about Changing SMS Construct Definitions: Changing existing constructs may not affect all objects associated with those constructs. Only those objects being stored or encountering a class transition after the construct definition is changed will be affected. A change to storage class or management class takes effect at the next storage management cycle only if the object needs management (such as class transition). For example, a change in the INITIAL ACCESS RESPONSE SECONDS parameter in a storage class may not cause any or all objects with that storage class to move within the storage hierarchy.

Monitoring DB2 Databases

You can use the following techniques to obtain information about performance and space allocation of DB2 databases, tables, and indexes that are used by OAM:

- DB2 RUNSTATS utility
- DB2 STOSPACE utility
- SQL statements

DB2 RUNSTATS Utility

RUNSTATS is a DB2 utility that scans a table space or indexes to gather information about space utilization and index efficiency. The information gathered is stored in the DB2 system tables and used by the SQL optimizer to select the best access paths during the bind process.

Run RUNSTATS to help evaluate the design of the database and determine when the REORG utility should be run for specific table spaces or indexes.

The output from RUNSTATS consists of DB2 updates to any or all of the following tables, depending on whether RUNSTATS was executed for a table space, indexes, or both:

- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSINDEXES
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSTABLEPART
- SYSIBM.SYSINDEXPART

By doing a global SELECT on the SYSIBM tables that were updated by RUNSTATS, you can determine what action, if any, should be taken to improve the performance of the system.

Attention:

1. Use caution. When RUNSTATS is active, no requests can be made to the affected tables.
2. After using RUNSTATS, rebind application plans that use the tables or indexes that were the subject of the RUNSTATS. This allows the DB2 optimizer to take advantage of new information about the structure of indexes.
3. After rebinding, examine the PLAN_TABLE output from the SQL EXPLAIN statement to ensure that all indexes are used. If PLAN_TABLE indicates that indexes are *not* used, override the DB2 catalog statistics, using the procedure specified in “Improving Data Access” in *DB2 for OS/390 Administration Guide*.

DB2 STOSPACE Utility

The DB2 STOSPACE utility determines the amount of space allocated for DB2 storage groups and their related table spaces and indexes. The utility updates DB2 system tables with the information it gathers.

Output from this utility consists of DB2 updates to the following tables:

- SYSIBM.SYSINDEXES
- SYSIBM.SYSTABLESPACE
- SYSIBM.SYSSTOGROUP

After STOSPACE execution is complete, use an SQL SELECT to view the tables that STOSPACE changed.

For RUNSTATS and STOSPACE utility syntax and usage notes, refer to *DB2 Utility Guide and Reference*.

SQL Statements

SQL statements can be used to determine the contents of various tables. Refer to *DB2 SQL Reference* for a complete listing of the DB2 system tables and what they contain. You should be familiar with these tables to learn about the OAM databases as they relate to DB2.

Tuning OAM

Tuning OAM is largely a matter of tuning its various components. Remember that application design, although not under the control of the storage administration team, plays a significant role in OAM performance and efficiency.

Tuning OAM Connections to DB2

When tuning OAM, there are a finite number of connections to DB2 from a batch environment that you need to consider. A number of functions, which are initiated by operator commands or automatically (for example, storage management cycle), may each result in multiple connections to DB2. OAM also establishes connections to process application requests. All of these connections established by OAM are in addition to the other necessary batch connections on your system. The total of all these connections at any given point in time must not exceed the DB2 limit because exceeding the limit causes OAM processing to fail.

The amount of concurrent function requests made of OAM control the tuning of OAM connections to DB2. In general, this involves limiting the number of concurrent functions requested automatically or by operator commands. A storage management cycle, for example, may establish three, four, or more connections to DB2 that persist for a good portion of the processing for each Object storage group. This must be taken into account when deriving a value for MAXS, as it controls the number of storage groups that the storage management cycle processes concurrently. While other calculations may seem to accommodate a larger number for MAXS, you must remember the DB2 limitation and adjust MAXS accordingly. Each installation is unique and must be tuned independently based upon actual experience; however, as a general guideline, as MAXS is increased above 10, the effectiveness of concurrency is diminished and may severely constrain processing in OAM or cause OAM processing to be unsuccessful. In an OAMplex, contention can increase with DB2 data sharing. When working with this type of environment, consider all the OAMs within the OAMplex when determining the storage group processing cycles and MAXS values for each instance of OAM.

Tuning the DB2 Databases

It is important to run the DB2 utility RUNSTATS on all of the databases after a significant number of objects are stored and volumes are defined. This is likely to decrease the length of the DB2 instruction path and to improve performance.

Performance is generally improved when DB2 uses an index to locate an object or object directory entry in a DB2 table. The index scan access path provides more direct access to the data than the table scan access path.

For example, if the DB2 utility RUNSTATS is run on a storage group with only one collection-name and object-name or only one collection-name and pending-action-date, DB2 can choose the table scan access path for operations such as OSREQ DELETE of an object and OSMC object processing. On the other hand, if the DB2 utility RUNSTATS is run after there are a significant number of objects in the Object storage group, the index scan access path can be chosen by DB2.

After running DB2 utility RUNSTATS, rebind the OAM application plans.

OAM databases can use the following facilities:
- REORG utility—reorganizes table spaces and indexes
- DB2 trace facilities—report on various internal system events
- Index, table space, and buffer pool tuning options—allow control of performance-related factors
- Concurrency control mechanisms (locks)—can be manipulated to increase concurrency or to improve performance

DB2 performance can be significantly affected by providing channel separation and DASD device separation when allocating DB2 logs, the directory, and object databases for each Object storage group. Resource Measurement Facility (RMF) can be used to monitor channel and device activity for tuning database and log allocation.

For more information, refer to *DB2 for OS/390 Administration Guide* and *RMF User's Guide*.

Table spaces are created with primary and secondary space. The secondary space is used when there is no more primary space. The secondary space is allocated from the DB2 storage group containing the primary space.

You must monitor the extension of table spaces into the secondary allocation to determine when to reorganize the individual database table spaces. It may be necessary to add volumes to the DB2 storage group so that additional extensions of the table spaces can occur.

Segmented Table Spaces

OAM specifies segmented table spaces in sample jobs and instructions for creating tables. This takes advantage of capabilities of segmented table spaces when doing INSERTs. The space maps for segmented table spaces provide a "guaranteed space" capability to find space in a table. Partitioned and simple table spaces lack this ability; however, there are instances where partitioned table spaces have definite advantages (see "Partitioning Table Spaces" on page 172 for more information).

OAM stores objects on DASD in DB2 tables. In most environments, there is a relatively static quantity of data stored overall. New objects are usually stored in both a time-sequenced manner and with object names which cause inserts to occur in timestamp order. Data is constantly being deleted from DASD as it is moved by class transition to optical or tape, or simply expiring. In general, large blocks of space become free in the object tables during each OSMC cycle. This space is best reused when segmented table spaces are used as DB2 does not have to search on a target page to determine if there is space prior to insertion. The space maps are able to pinpoint available and sufficient space. This permits managing tables at a predetermined size with greatly reduced maintenance.

Some installations experience significantly longer processing time when segmented table spaces are not used which causes increased search activity and more time consuming searches for available space. This reduced performance, primarily when using simple table spaces, can only be managed by constant management of the table space (reorganization activity and reallocation) to assure that there is always both space in the current extent of the data set supporting a DB2 table and sufficient extents available to guarantee space for storing objects. The use of segmented table spaces reduces the need for such manual management.

Note: Simple table spaces are not recommended for use with OAM object tables or object directory tables. Space searches and space reuse in simple table spaces consume a much greater proportion of processing time.

Partitioning Table Spaces

Partitioned table spaces permit large tables to be split into smaller entities which are managed more easily using DB2 utilities. Operations such as IMAGE COPY and REORG are more efficient and consume less total aggregate processing time when performed on smaller entities when tables are larger than 2 GB.

Partitioned table spaces are recommended when:
- Tables become very large
- Data may be relatively static for long periods of time
- DB2 maintenance must be minimized
- Any combination of these reasons

Backup and recovery actions for DB2 tables/table spaces are necessary under all circumstances. Regular IMAGE COPY operations and proper safeguard of DB2 logging is necessary to provide contingency for outages of any type.

Reorganizing tables is a different matter. Under circumstances where an object table or object directory table can be managed at a stable total allocation, segmented table spaces nearly eliminate any need to reorganize tables using the REORG operation. OAM uses DB2 indexes for all SELECTs and INSERTs as a consequence of its underlying design. The use of indexes removes the requirement for the tables to be in strict cluster index order. When a table is relatively new and is loaded with data, the RUNSTATS utility should be used to be certain that DB2 has good information on the order within the table in its catalog tables. Following RUNSTATS, a BIND with the EXPLAIN parameter should be performed to determine if DB2 is using the indexes. After this initial use of RUNSTATS, avoid the further use of the RUNSTATS utility. Over time with deletes of older objects and reuse of space for new objects, the object directory and object tables tend not to be in strict cluster sequence. It is not important that OAM object and object directory tables be in cluster sequence and regularly reorganized. OAM access to data is entirely though DB2 indexes. The initial “decision” by DB2 to use indexes when a table is created will be maintained, and indexes will be used for access, as long as RUNSTATS utility is not used when tables are not maintained in strict cluster index sequence. The use of RUNSTATS without reorganizing a table could result in DB2 discontinuing use of indexes.

The advantages described here are best used when segmented table spaces are used for objects and object directory entries. As stated, simple table spaces should not be used for OAM. There are circumstances when the INSERT performance differences between segmented and partitioned table spaces are not as important as minimizing the work load of DB2 maintenance activity. It is the decision of the installation whether to accept less possible performance and use partitioned table spaces based on their unique operating circumstances.

Partitioning Object Storage Tables

Because of the large amount of data that can be stored in the OAM Object Storage Tables (the 4K Object Storage Table and the 32K Object Storage Table) associated with each Object storage group, you may choose to partition the DB2 table spaces containing each of these two tables. Both the 4K Object Storage Table and the 32K Object Storage Table are stored in separate DB2 table spaces. You may partition the DB2 table space containing the 4K Object Storage Table or the DB2 table space containing the 32K Object Storage Table, or both. For information about the advantages and disadvantages of partitioned table spaces, refer to *DB2 for OS/390 Administration Guide*.

Partitioning the 4K Object Storage Tables

During OAM installation and customization (using the default SAMPLIB members provided with the product), OAM creates a nonpartitioned unique clustered index on the 4K Object Storage Table using a composite key consisting of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

You may change the DB2 table space containing the 4K Object Storage Table into a partitioned table space. If you do so, decide which column or columns on the 4K Object Storage Table to use for the partitioning key. The following two examples of columns in the 4K Object Storage Table may be used for the partitioning key:

- Collection ID column (OTCLID)
- Object name column (OTNAME)

OAM does not update the value of the collection ID column (OTCLID) in a row in the 4K Object Storage Table, so this column may be used in a partitioning key. OAM does not update the value of the object name column (OTNAME) in a row in the 4K Object Storage Table, so this column may be used in a partitioning key. If you use the object name column (OTNAME) as the partitioning key, remember that DB2 only uses the first 40 bytes of the partitioning key to actually partition the data.

In order to create the DB2 table space containing the 4K Object Storage Table as a partitioned table space, you must modify the CREATE TABLESPACE SQL statements by adding a NUMPARTS clause for the HLQ.OSMOTS04 table spaces in the CBRISQL0 job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4K Object Storage Table as a partitioned table space, you must define a partitioned index on the 4K Object Storage Table. The partitioned index can be created by adding a CREATE INDEX SQL statement to the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4K Object Storage Table as a partitioned table space, the partitioned index must also be the clustering index. Therefore, the default index that OAM creates on the 4K Object Storage Table (HLQ.OBJT04X1) cannot be a clustering index. In this case, you must change the default index that OAM creates on the 4K Object Storage Table (HLQ.OBJT04X1) to a nonclustered index by removing the CLUSTER keyword from the CREATE INDEX SQL statement for the HLQ.OBJT04X1 index in the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 4K Object Storage Table as a partitioned table space, there must still be a unique nonpartitioned index on the composite key in order for OAM to function properly that consists of both:

- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

Note: Having a partitioned index and a nonpartitioned index on the 4K Object Storage Table may diminish some of the benefits of partitioning the 4K Object Storage Table in the first place.

In addition to changing the SQL statements contained in the CBRISQL0 sample jobs in SYS1.SAMPLIB, also update the CBRIALC0 job in SYS1.SAMPLIB. Include Access Method Services (AMS) DEFINE CLUSTER commands to preallocate a VSAM linear data set (LDS) for each of the partitions that you plan on having for each partitioned table space containing the 4K Object Storage Table associated with each Object storage group. Also use the AMS DEFINE CLUSTER command to preallocate a VSAM linear data set for each partition comprising the partitioned index that you plan to create. The data set names associated with each VSAM linear data set must conform to DB2 data set naming conventions as specified in the *DB2 for OS/390 Administration Guide*.

Keep in mind that DB2 free space search algorithms are not as efficient for partitioned table spaces as they are for segmented table spaces. As a result of partitioning the DB2 table spaces containing the OAM 4K Object Storage Tables, you may be impacting the performance when small objects are being stored to DASD.

By partitioning the DB2 table space containing the OAM 4K Object Storage Table, you are accepting the following responsibilities:
- That OSREQ STORE performance, when storing small objects to DASD, may not be as fast as when using a segmented table space for the OAM 4K Object Storage Table
- That OSMC transition of small objects to DASD may not be as fast as when using a segmented table space for the OAM 4K Object Storage Table
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned table space
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned index

Partitioning the 32K Object Storage Tables

During OAM installation and customization (using the default SAMPLIB members provided with the product), OAM creates a nonpartitioned unique clustered index on the 32K Object Storage Table using a composite key consisting of both:
- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order

You may change the DB2 table space containing the 32K Object Storage Table into a partitioned table space. If you do so, you need to decide what column or columns on the 32K Object Storage Table to use for the partitioning key. The following two examples of columns in the 32K Object Storage Table may be used for the partitioning key:
- Collection ID column (OTCLID) or
- Object name column (OTNAME)

Because OAM does not update the value of the collection ID column (OTCLID) in a row in the 32K Object Storage Table, this column may be used in a partitioning key. OAM does not update the value of the object name column (OTNAME) in a row in the 32K Object Storage Table, so this column may be used in a partitioning key. If you use the object name column (OTNAME) as the partitioning key, remember that DB2 only uses the first 40 bytes of the partitioning key to actually partition the data.

In order to create the DB2 table space containing the 32K Object Storage Table as partitioned table space you must modify the CREATE TABLESPACE SQL statements by adding a NUMPARTS clause for the HLQ.OSMOTS32 table spaces in the CBRISQL0 job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32K Object Storage Table as a partitioned table space, you must define a partitioned index on the 32K Object Storage Table. The partitioned index can be created by adding a CREATE INDEX SQL statement to the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32K Object Storage Table as a partitioned table space, the partitioned index must also be the clustering index. Therefore, the default index that OAM creates on the 32K Object Storage Table (HLQ.OBJT32X1) cannot be a clustering index. In this case, you must change the default index that OAM creates on the 32K Object Storage Table (HLQ.OBJT32X1) to a nonclustered index by removing the CLUSTER keyword from the CREATE INDEX SQL statement for the HLQ.OBJT32X1 index in the CBRISQL0 sample job in SYS1.SAMPLIB.

If you create the DB2 table space containing the 32K Object Storage Table as a partitioned table space, there must still be a unique nonpartitioned index on the composite key in order for OAM to function properly that consists of the following:
- Collection ID column (OTCLID) in ascending order
- Object name column (OTNAME) in ascending order
- Object segment (OTSEG) in ascending order

Note: Having a partitioned index and a nonpartitioned index on the 32K Object Storage Table may diminish some of the benefits of partitioning the 32K Object Storage Table in the first place.

In addition to changing the SQL statements contained in the CBRISQL0 sample job in SYS1.SAMPLIB, also update the CBRIALC0 job in SYS1.SAMPLIB. Include Access Method Services (AMS) DEFINE CLUSTER commands to preallocate a VSAM linear data set (LDS) for each of the partitions that you plan on having for each 32K Object Storage Table associated with each Object storage group. Also use the AMS DEFINE CLUSTER command to preallocate a VSAM linear data set for each partition comprising the partitioned index that you plan to create. The data set names associated with each VSAM linear data set must conform to DB2 data set naming conventions as specified in the *DB2 for OS/390 Administration Guide*.

Keep in mind that DB2 free space search algorithms are not as efficient for partitioned table spaces as they are for segmented table spaces. As a result of partitioning the DB2 table spaces containing the OAM 32K Object Storage Tables, you may be impacting the performance when large objects are being stored to DASD.

By partitioning the DB2 table space containing the OAM 32K Object Storage Table, you are accepting the following responsibilities:
- That OSREQ STORE performance when storing large objects to DASD may not be as fast as when using a segmented table space for the OAM 32K Object Storage Table
- That OSMC transition of large objects to DASD may not be as fast as when using a segmented table space for the OAM 32K Object Storage Table
- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned table space

- To create and manage each of the underlying VSAM linear data sets associated with each partition of the partitioned index

Tuning Object Retrieval Response Time

The OAM response time for retrieval of objects may be a key factor in the performance of your application, especially if the application is intended for interactive use. The minimum time to retrieve an object from an unmounted cartridge in an optical library is typically 15 to 30 seconds. These times increase when the resource is attached to a different OAM in an OAMplex and cross-system processing is required. Understand the retrieval response time requirements for your applications and monitor the actual response time achieved.

The key to providing the required response time is to assign objects to storage classes that have an adequate INITIAL ACCESS RESPONSE SECONDS value defined and to management classes that do not cause a transition to a slower storage class until the frequency of retrieving the objects is reasonably low. The primary attributes of a management class that can be used to control transition are TIME SINCE CREATION and TIME SINCE LAST USE.

Note: Do not use TIME SINCE LAST USE if the UPD=N option is used on the OAM1,CBRINIT statement in IEFSSN*xx* PARMLIB member.

Use the OSREQ QUERY function to obtain the estimated retrieval response time for an object. The OSREQ QUERY function also provides descriptive information concerning objects residing on the following storage media:
- Direct access storage device (DASD)
- An optical disk volume inside of an optical library
- An optical disk volume on the shelf
- A tape volume inside an IBM automated tape library dataserver
- A shelf-resident tape volume (a tape volume that resides outside an IBM automated tape library dataserver)

OAM returns this descriptive information, along with the primary, backup, and secondary backup retrieval order keys, in the Query Element List. QUERY searches the object directory for a match on the specific name in the NAME keyword and returns a single query element. A generic search for each object whose name matches the partially qualified name specified in the NAME keyword is also supported, and returns a query element for each object found. The output of a QUERY may be used as the input to an OSREQ RETRIEVE request. For more information concerning the Query Element List, refer to *z/OS DFSMS OAM Application Programmer's Reference*.

Additionally, for library-resident objects whose retrieval can be predicted in advance, you may want to fetch the objects before they are needed for interactive use. This can be done by performing the following activities:
- Using the OSREQ macro to change objects' storage classes to storage classes having a nonzero value for INITIAL ACCESS RESPONSE SECONDS (stored on removable media—tape or optical)
- Running the storage management cycle to move the objects to the proper level of the object storage hierarchy

Note: An automated way of prefetching objects when retrieval of objects is regular (such as the first day of each month) is to assign the objects to a management class that causes a transition to a fast storage class just prior to using the objects. This is controlled by using the PERIODIC parameter of the management class. See page 145 for a discussion of the attributes for this parameter.

See “Balancing Library Usage” on page 178 for a discussion of how drive availability between libraries can affect object retrieval response time.

Tuning the Storage Management Cycle

The storage management cycle moves objects between DASD, optical, and tape media, writes backup copies of objects, deletes expired objects, and expires optical cartridges when all the optical cartridge expiration criteria have been set. It should be run when the application work load is at a minimum.

You can obtain the shortest storage management cycle by making the best use of the library drives and stand-alone or operator-accessible drives in your SMS configuration. Your intent should be to process as many Object storage groups concurrently and use as many drives concurrently as possible without introducing contention for drives by different Object storage groups (which causes unnecessary mounts and demounts of cartridges).

OAM provides the following controls for tuning the storage management cycle:

- The DRIVE STARTUP threshold attribute of each Object storage group definition and each Object Backup storage group definition determines the number of optical drives that are used concurrently for that storage group.
- The TAPEDRIVESTARTUP threshold optional subparameter of the STORAGEGROUP parameter on the SETOAM statement, determines when OAM is to start additional tape drives for writing object data to tape volumes belonging to the Object and Object Backup storage groups. Consider the MAXTAPESTORETASKS keyword (for the OAM global level and at the storage group level) in conjunction with TAPEDRIVESTARTUP threshold for further controls within the storage management cycle.

  Note: Do not specify a number greater than the number of tape drives available to OAM for the MAXTAPESTORETASKS or the MAXTAPERETRIEVTASKS subparameters. This can cause a system to go into allocation recovery and attempt to allocate tape drives after all tape drives are in use causing system problems.
- The CYCLE START TIME and CYCLE END TIME attributes of each Object storage group control the window in which the storage management cycle begins processing the storage group.
- The MAXS parameter of the OAM cataloged procedure controls the number of storage groups that the storage management cycle processes concurrently.
- The UPD=N option on the OAM1 statement in IEFSSN*xx* member of PARMLIB can be used to reduce unnecessary retrieval and update of objects’ directory entries during the OSMC cycle if your installation’s management classes do not use the TIME SINCE LAST USE or EXPIRE AFTER DAYS USAGE parameters.
- In an OAMplex, the OSMC processing system name attribute of each Object storage group definition controls where OSMC processing is done for that storage group. Using this parameter and separating hardware between storage groups can balance workload across systems for OSMC processing. Localizing hardware and highest usage will reduce XCF overhead.

If your processing includes making object backups on stand-alone or operator-accessible optical disk drives, run as many Object storage groups concurrently as there are usable stand-alone or operator-accessible optical drives. If backup copies are not being made, you may be able to run as many Object storage groups concurrently as there are usable library drives in the configuration. Remember that in determining the number of Object storage groups to run concurrently, there are DB2 limitations which must be taken into account. See “Tuning OAM Connections to DB2” on page 170 for additional information on these DB2 limitations.

Remember that OSMC functions other than the storage management cycle you start (for example, Volume Recovery utility, Move Volume utility, and others) are consumers of resources as well and need to be considered in your usage of the MAXS and DRIVE STARTUP threshold controls.

To avoid contention for drives within any one library, analyze the windows for processing each Object storage group. For example, if volumes for eight storage groups all reside in the same single optical library with four drives and MAXS=10, then the windows for the groups should be set so that no more than four overlap at any given time.

Note: Consider all OAMs within an OAMplex when making decisions that affect those resources.

CAUTION:
If you plan on using the CBRHADUX installation exit that is shipped with the SAMPLIB or plan on editing or creating your own CBRHADUX installation exit that does not allow expiration of objects, then this can cause OSMC performance problems if you have not properly established your expiration criteria in your Storage Management Subsystem (SMS) management class constructs.

If you do not plan on expiring objects and have established your CBRHADUX installation exit to return with an indication that no expiration is allowed, you must make sure that your SMS Management Class has expiration criteria that will not cause OSMC to continually pick objects to be expired. Always be sure that your management class sets the expiration criteria to NEVER expire if you do not plan to expire OAM objects.

Balancing Library Usage

OAM attempts to balance drive use within a given library; however, in a system with multiple libraries, the level of activity can vary greatly among libraries. Activity can be affected by the assignment of Object or Object Backup storage groups to libraries, the frequency of retrieval of objects from optical or tape cartridges, and the availability of scratch cartridges.

If one library has much more activity than the others, the response time for work on that library can be lengthened. You should monitor the number of cartridge mounts in each library by examining the console log.

To balance the work load, reassign Object or Object Backup storage groups to other libraries, and eject and move the cartridges to the corresponding libraries. Also, for an Object storage group that is defined to be resident in multiple libraries, cartridges in the Object storage group can be ejected from the overly active library and entered into another library assigned to the Object storage group.

When determining which volume to use for a write request, OAM attempts to find the volume that is available to the storage group and that has the least amount of free space, but enough space for the object that is to be written. If a new library is added to a configuration and the new library is added to an existing Object or Object Backup storage group definition, it is a good idea to move some of the partially used volumes that belong to the Object or Object Backup storage group instead of populating a new library only with scratch or unused grouped volumes. This will help to distribute the read and write workload across the libraries.

Using Appropriate Transport Classes within XCF

In an OAM Parallel Sysplex environment, cross coupling facility (XCF) message services are used to send requests and data between instances of OAM within an OAMplex. See “OAMplex” on page 3 for more information concerning an OAMplex and refer to *z/OS Parallel Sysplex Overview* for more information concerning a Parallel Sysplex. XCF messaging services, a function within XCF, uses transport classes to send XCF messages and data through the coupling facility between the various systems within a sysplex environment.

Transport classes are used by XCF messages service to group messages that are to be sent between systems within a Parallel Sysplex. Messages are assigned to transport classes based on the group name (defining specific transport classes to OAM, for example), the message size, or both. Each transport class has its own resources that consist of buffers and one or more inbound and outbound paths. In most cases, it is more efficient to pool the resources and have the transport classes based on message size, rather than on group name.

XCF message buffers are managed by correctly selecting the size of the messages most frequently sent from specific buffer pools and by specifying an adequate upper limit for the size of the buffer pool. There are multiple default transport classes of various sizes assigned to the multiple buffers in the coupling facility. XCF determines which transport class is to be used depending on the size of the message or object in the buffer being transported.

XCF attempts to optimize the use of transport classes by selecting a class that is large enough to handle all the data being transported. For example, if there is a message or data being sent from one instance of OAM to another OAM system within an OAMplex through the coupling facility using XCF message services that is 5K in size, XCF might use the 5K default transport class as the vehicle to send the message. XCF tries to select the best fit transport class for the buffer size. However, XCF might also choose the 5K transport class to handle a 2K message if that is the best fit available at the time. The 5K default transport would be large enough to handle the request; however, the buffer is not being used efficiently.

It is possible to create customized transport classes based on message size (or object size) to use specifically for your own data by defining transport classes based on message or object size and assigning them based on the OAM XCF group name along with the default transport classes. Defining your own transport classes allows you to determine the best fit for your objects to optimize the use of the transport class for your group class buffer size.

Note: Customizing your own transport classes works best if your installation has standard object sizes. Additionally, you would want to create smaller transport classes for OAM to handle the smaller messages used to communicate configuration updates. In this case, you would have the best fit transport classes for your data and messages.

Perform the following to determine if there is a legitimate need for user defined transport classes:

1. Use the XCF default transport classes assigned to the buffers first to determine if they sufficiently accommodate the size of your data and are being used in an efficient manner.
2. Modify your configuration (storage group, library, and drive definitions) to best utilize your resources and reduce unnecessary XCF messaging for processing transactions. Some libraries span several storage groups, which may increase the need to send messages using XCF. Update your configuration to minimize the frequency and the amount of data that needs to be transported. For OSMC processing, try to run OSMC on the OAM that is managing and controlling the hardware associated with both the Object storage group being processed and the Object Backup storage group. If different storage groups are being processed on multiple OAMs concurrently and backup copies of objects need to be written, you should have hardware available to the Object Backup storage groups on each OAM that is doing the processing.
3. Run RMF reports for XCF activity (refer to *RMF User's Guide*) and analyze the reports to determine which default transport classes are used with what buffer size and how often the transport classes are used.

Note: If your average object size is larger, consider using a CTC direct connection for transporting XCF messages and data to improve performance.

If you determine, after you have performed the previous steps, that there is sufficient need to define specific transport classes to the OAMplex to optimize the use of system resources, defined transport classes can be used along with the XCF defaults. For information on how to calculate message buffer space, refer to *z/OS MVS Setting Up a Sysplex*.

Measuring OAM Transaction Performance Using SMF

OAM uses system management facility (SMF) recording for gathering OAM statistical information to allow customers to measure the performance of OAM at the application programming interface level (the OSREQ macro interface). Refer to *z/OS DFSMS OAM Application Programmer's Reference* for more information regarding the OSREQ macro. Also, see "Appendix D. OAM System Management Facility (SMF) Records" on page 475 for more detailed information of SMF.

The OAM SMF record allows the collection of statistical information about OAM usage for planning and diagnosis purposes such as:

- Information system usage accounting and charge back to end user departments
- Performance analysis and monitoring to make certain that their information systems are as finely tuned as possible
- Capacity planning to determine when to procure additional hardware resources, such as storage devices (DASD, tape, optical) and media
- Potential problem determination data

The OAM SMF record supports subtypes, which are assigned to almost all OAM activities. The MVS system operator or system programmer can dynamically select the OAM SMF record subtypes to be recorded. The following activities have associated subtype records:

- Invocations of OSR functions:
  - ACCESS
  - CHANGE

- DELETE
- QUERY
- RETRIEVE
- STORE
- UNACCESS

- Invocations of OSMC storage management activities:
  - Storage group processing
  - DASD space management processing
  - Volume recovery utility
  - Single object recovery utility
  - Library space management
  - Move volume utility
- Library control system (LCS) optical library activities
  - Optical library varies online
  - Optical library varies offline
  - Optical drive varies online
  - Optical drive varies offline
  - Optical cartridge entry
  - Optical cartridge eject
  - Optical cartridge label
  - Optical cartridge audit
  - Optical cartridge mount
  - Optical cartridge demount
  - Optical write
  - Optical read
  - Optical logical delete
  - Optical physical delete
- LCS object tape activities:
  - Object tape write
  - Object tape read
  - Object tape demount
  - Object tape logical delete

OAM SMF Record Subtypes

OAM records SMF records in the SMF data sets to account for OAM activity. The OAM SMF record is a type 85 (X'55'). The OAM SMF record supports record subtypes and begins with a standard 48-byte SMF record header. Each OAM SMF record contains three sections:

- Standard 48-byte SMF record header
- 112-byte OAM product section
- Variable length OAM data section

Table 29 lists the OAM SMF record subtypes.

Table 29. Record Subtypes and Descriptions

Record Subtype	Record Size	Description
1	324	OSREQ Access
2	324	OSREQ Store
3	324	OSREQ Retrieve
4	324	OSREQ Query
5	324	OSREQ Change
6	324	OSREQ Delete

Table 29. Record Subtypes and Descriptions (continued)

Record Subtype	Record Size	Description
7	324	OSREQ Unaccess
32	336	OSMC Storage Group Processing
33	336	OSMC DASD Space Management
34	336	OSMC Volume Recovery Utility
35	336	OSMC Move Volume Utility
36	296	OSMC Single Object Recovery Utility
37	184	OSMC Library Space Management
64	256	LCS Optical Drive Vary Online
65	256	LCS Optical Drive Vary Offline
66	256	LCS Optical Library Vary Online
67	256	LCS Optical Library Vary Offline
68	284	LCS Optical Cartridge Entry
69	284	LCS Optical Cartridge Eject
70	284	LCS Optical Cartridge Label
71	284	LCS Optical Volume Audit
72	284	LCS Optical Volume Mount
73	284	LCS Optical Volume Demount
74	variable (min = 416, max = 32 744)	LCS Optical Write Request
75	416	LCS Optical Read Request
76	380	LCS Optical Logical Delete Request
77	variable (min = 380, max = 32 744)	LCS Optical Physical Delete Request
78	variable (min = 380, max = 32 744)	LCS Object Tape Write Request
79	416	LCS Object Tape Read Request
87	228	LCS Object Tape Volume Demount (OAM usage)
88	380	LCS Object Tape Logical Delete Request

OAM SMF Start and End Time Accuracy

Each OAM SMF record has a function start and end time in the common OAM product section. The start and end times are in fields R85PSTRT and R85PENDT respectively, and they are in System/390 STORE CLOCK (STCK) instruction format.

The starting time of the OAM function is obtained as soon as possible so that the elapsed time of the function includes as much OAM processing time as possible.

The ending time of the OAM function is obtained as soon as possible to the end of the function so that the elapsed time of the function includes as much OAM processing time as possible. However, since the function end time (in field R85PENDT) is set and the elapsed time (in field R85PRESP) is calculated prior to passing the record to SMF for recording, the elapsed time does not include the time required to invoke SMF and the time required to copy the record into the SMF buffers.

Identifying OAM Transaction Activity Using RMF

A tool that can be used to assist you in analyzing your business environment if you currently are involved in object support is the Resource Measurement Facility (RMF). OAM provides RMF to allow the installation to obtain reports on workload and transaction activity against specific report performance groups that are defined in the IEAICS*xx* PARMLIB member. These reports provide an installation with the ability to monitor and obtain a quick snapshot of OAM's performance at a given time. The RMF transaction reporting enhancement assists customers using OAM for a variety of applications in performance monitoring, analysis, and tuning.

There are two report types obtainable from RMF:
- RMF Monitor I Workload Activity Report
- RMF Monitor II Transaction Activity Report

Note: For complete information about the Resource Measurement Facility, refer to *RMF User's Guide*.

OAM Transaction Classes

OAM passes a transaction completion message to the MVS/System Resource Manager (SRM) containing the following information:
- A subsystem name of "OAM"
- A transaction class name (OSREQ or OSMC)
- A transaction name (also referred to as report performance group)
- The transaction start time in S/390 stock clock (STCK) instruction format

There are two OAM transaction classes used to support RMF:

OSREQ The OSREQ transaction class represents invocation of the OSREQ macro (the programming interface provided by OAM).

OSMC The OSMC transaction class represents activities performed by the OAM Storage Management Component (OSMC).

Table 30 describes the transaction names associated with the OSREQ transaction class.

Table 30. OSREQ Transaction Names

Transaction Name	Description
ACCESS	Represents completion of an OSREQ ACCESS macro invocation.
CHANGE	Represents completion of an OSREQ CHANGE macro invocation.
DELETED	Represents completion of an OSREQ DELETE macro invocation.
DELETEDB	Represents completion of an OSREQ DELETE macro invocation where the primary copy of the object resided on DASD and a backup copy of the object was also deleted during the OSREQ DELETE processing.
DELETEO	Represents completion of an OSREQ DELETE macro invocation where the primary copy of the object resided on optical.

Table 30. OSREQ Transaction Names (continued)

Transaction Name	Description
DELETEOB	Represents completion of an OSREQ DELETE macro invocation where the primary copy of the object resided on optical and a backup copy of the object was also deleted during the OSREQ DELETE processing.
DELETET	Represents completion of an OSREQ DELETE macro invocation where the primary copy of the object resided on tape.
DELETETB	Represents completion of an OSREQ DELETE macro invocation where the primary copy of the object resided on tape and a backup copy of the object was also deleted during the OSREQ DELETE processing.
QUERY	Represents completion of an OSREQ QUERY macro invocation where the object name was a fully-qualified object name. Objects on DASD, optical, or tape can be queried.
QUERYG	Represents completion of an OSREQ QUERY macro invocation where the function performed was a generic query as a result of a partially qualified object name (one containing an asterisk in the low-level qualifier) being specified on the OSREQ QUERY macro.
RETRVBO	Represents completion of an OSREQ RETRIEVE request where the backup copy of the object was retrieved from optical due to the VIEW=BACKUP1 (for the first backup copy) or VIEW=BACKUP2 (for the second backup copy) keyword being specified on the OSREQ RETRIEVE macro.
RETRVBT	Represents completion of an OSREQ RETRIEVE request where the backup copy of the object was retrieved from tape due to the VIEW=BACKUP1 (for the first backup copy) or VIEW=BACKUP2 (for the second backup copy) keyword being specified on the OSREQ RETRIEVE macro.
RETRVPD	Represents completion of an OSREQ RETRIEVE request where the primary copy of the object was retrieved from DASD.
RETRVPO	Represents completion of an OSREQ RETRIEVE request where the primary copy of the object was retrieved from optical.
RETRVPOO	Represents completion of an OSREQ RETRIEVE request where the following conditions exist: • The primary copy resides on an optical disk volume. • The optical disk volume that the primary copy resides on is unreadable, in an offline library, or in a nonoperational library. • A backup copy of the object exists. • A backup copy of the object resides on an optical volume. • The Access Backup facility is active. • A backup copy of the object was retrieved from the optical volume and returned to the application program.

Table 30. OSREQ Transaction Names (continued)

Transaction Name	Description
RETRVPOT	Represents completion of an OSREQ RETRIEVE request where the following conditions exist: • The primary copy resides on an optical disk volume. • The optical disk volume that the primary copy resides on is unreadable, in an offline library, or in a nonoperational library. • A backup copy of the object exists. • A backup copy of the object resides on a tape volume. • The Access Backup facility is active. • A backup copy of the object was retrieved from the tape volume and returned to the application program.
RETRVPT	Represents completion of an OSREQ RETRIEVE request where the primary copy of the object was retrieved from tape.
RETRVPTO	Represents completion of an OSREQ RETRIEVE request where the following conditions exist: • The primary copy resides on a tape volume. • The tape volume that the primary copy resides on is unreadable, in an offline library, or in a nonoperational library. • A backup copy of the object exists. • A backup copy of the object resides on an optical volume. • The Access Backup facility is active. • A backup copy of the object was retrieved from the optical volume and returned to the application program.
RETRVPTT	Represents completion of an OSREQ RETRIEVE request where the following conditions exist: • The primary copy resides on a tape volume. • The tape volume that the primary copy resides on is unreadable, in an offline library, or in a nonoperational library. • A backup copy of the object exists. • A backup copy of the object resides on a tape volume. • The Access Backup facility is active. • A backup copy of the object was retrieved from the tape volume and returned to the application program.
STORDASD	Represents completion of an OSREQ STORE macro invocation where the object was stored on DASD.
STOROPT	Represents completion of an OSREQ STORE macro invocation where the object was stored on optical.
STORTAPE	Represents completion of an OSREQ STORE macro invocation where the object was stored on tape.
UNACCESS	Represents completion of an OSREQ UNACCESS macro invocation.

Table 31 describes the transaction names associated with the OSMC transaction class.

Table 31. OSMC Transaction Names

Transaction Name	Description
DASDREAD	Represents completion of reading an object by way of an SQL SELECT statement from either the 4K or 32K Table by OSMC during the OSMC storage management cycle or the OSMC Volume Recovery utility.

Table 31. OSMC Transaction Names (continued)

Transaction Name	Description
DASDWRIT	Represents completion of writing an object by way of an SQL INSERT statement to either the 4K or 32K Table by OSMC during the OSMC storage management cycle.
DIRDEL	Represents completion of deleting a row by way of an SQL DELETE statement from the Object Directory Table during the OSMC storage management cycle or the OSMC DASD space management function.
DIRUPD	Represents completion of updating a row in the Object Directory Table by OSMC during any one of the following OSMC functions: • OSMC storage management cycle • OSMC Volume Recovery utility • OSMC single object recovery utility • OSMC Move Volume utility
OBJEXPIR	Represents the expiration of an object by OSMC during OSMC storage management cycle or OSMC DASD space management facility.
OBJMOVE	Represents the completion of processing an object by the OAM Move Volume utility.
OBJPROC	Represents the completion of processing an object (other than expiration of the object) by OSMC during the OSMC storage management cycle.
OBJRECVS	Represents the completion of processing an object by the OAM single object recovery utility.
OBJRECVV	Represents the completion of processing an object by the Volume Recovery utility.

RMF Monitor I Workload Activity Report

In order for RMF to receive these messages from SRM and for a RMF Monitor I Workload Activity Report to be generated, the following criteria must be met:

- RMF must be initialized.
- An RMF Monitor I session must be active and collecting system workload activity.
- Report performance groups (transaction names associated with the different transaction class name) must be defined in the installation control specifications (ICS) located in the IEAICS*xx* PARMLIB member of PARMLIB.
- The ICS, containing the report performance groups for the OAM transaction names, must be activated by way of the *SET ICS=xx* command at the MVS system console.

Since OAM transaction names are defined as report performance groups in the ICS, and RMF only reports on report performance groups in the type 1 subreport (performance group period) and the type 2 subreport (performance group), then:

- The WKLD(PERIOD) RMF option must be in effect requesting reporting by *performance group period* (type 1 subreport) on the Workload Activity Report **or**,
- The WKLD(GROUP) RMF option must be in effect requesting reporting by *performance group* (type 2 subreport) on the Workload Activity Report.

The following fields appear on the Monitor I Workload Activity Report containing valid data for OAM transactions performed against the report performance groups that are defined in the ICS:

AVERAGE TRANSACTIONS
The average number of active transactions during the interval

MPL The average number of transactions resident in central storage during the interval

ENDED TRANS
The number of transactions that ended during the interval

AVG TRANS TIME
The average elapsed time of ended transactions

STD DEV
The standard deviation of the average elapsed time of ended transactions

| Figure 24 is an example of the Sample RMF Monitor I Workload Activity Report, which shows OAM transaction completions for various report performance groups.

```
                                         W O R K L O A D   A C T I V I T Y
                                                                                                                    PAGE    1
            MVS/ESA                 SYSTEM ID 3090            DATE 09/01/93            INTERVAL 12.26.657
            SP5.2.0                 RPT VERSION 5.2.0         TIME 11.15.26            IPS = IEAIPS04
         OPT = IEAOPT00                      SUMMARY BY PERFORMANCE GROUP   SERVICE DEFINITION COEFFICIENTS     SU/SEC=1162.3
         ICS = IEAICSTU                                                     IOC =  5.0  CPU = 10.0  SRB = 10.0  MSO =  3.0000
               TIME                        AVERAGE ABSORPTION,                                                   AVG TRANS. TIME,
PGN  PGP  DMN  SLICE  INTERVAL SERVICE  AVG TRX SERV RATE,     PAGE-IN RATES      STORAGE         TRANSACTIONS     STD. DEVIATION
               GROUP                    TCB+SRB SECONDS, %                                                           HHH.MM.SS.TTT
0000 ALL  ALL  ALL     IOC       1,170  ABSRPTN        11    SINGLE      0.02   AVERAGE  101.33   AVG       7.00   TRX 000.00.00.000
                       CPU       25423  TRX SERV       11    BLOCK       0.03                     MPL       7.00   SD  000.00.00.000
                       MSO       20602  TCB           2.1    HSP         0.00   TOTAL    709.80   ENDED        0
                       SRB       11766  SRB           1.0    HSP MISS    0.00   CENTRAL  709.80   END/SEC   0.00   QUE 000.00.00.000
                       TOT       58961  TCB+SRB%      0.4    EXP SNGL    0.00   EXPAND     0.00   #SWAPS       2
                       PER SEC      78                       EXP BLK     0.00                                      TOT 000.00.00.000
SUBSYS = OAM          TRXCLASS =            ACCTINFO = NO
USERID =               TRXNAME =
0001 ALL  ALL  ALL     IOC       23091  ABSRPTN         8    SINGLE      0.01   AVERAGE  142.10   AVG      22.99   TRX 000.00.00.000
                       CPU       31226  TRX SERV        8    BLOCK       0.07                     MPL      22.99   SD  000.00.00.000
                       MSO       74840  TCB           2.6    HSP         0.00   TOTAL   3,268.5   ENDED        0
                       SRB       10387  SRB           0.8    HSP MISS    0.00   CENTRAL 3,268.5   END/SEC   0.00   QUE 000.00.00.000
                       TOT      139.5K  TCB+SRB%      0.4    EXP SNGL    0.00   EXPAND     0.00   #SWAPS       0
                       PER SEC     186                       EXP BLK     0.00                                      TOT 000.00.00.000
0002 ALL  ALL  ALL     IOC       17730  ABSRPTN     1,577    SINGLE      2.48   AVERAGE  343.16   AVG       0.12   TRX 000.00.01.924
                       CPU       40034  TRX SERV    1,522    BLOCK       1.40                     MPL       0.12   SD  000.00.03.470
                       MSO       80126  TCB           3.4    HSP         0.00   TOTAL     41.46   ENDED       61
                       SRB       4,399  SRB           0.3    HSP MISS    0.00   CENTRAL   41.46   END/SEC   0.08   QUE 000.00.11.126
                       TOT      142.3K  TCB+SRB%      0.4    EXP SNGL    0.00   EXPAND     0.00   #SWAPS      86
                       PER SEC     190                       EXP BLK     0.00                                      TOT 000.00.13.050
0003 ALL  ALL  ALL     ZEROS
THRU
0099

SUBSYS = OAM          TRXCLASS = OSREQ      ACCTINFO = NO
USERID =               TRXNAME =
0400 ALL  ALL  ALL     IOC           0  ABSRPTN         0    SINGLE      0.00   AVERAGE    0.00   AVG       0.00   TRX 000.00.00.616
                       CPU           0  TRX SERV        0    BLOCK       0.00                     MPL       0.00   SD  000.00.01.248
                       MSO           0  TCB           0.0    HSP         0.00   TOTAL      0.00   ENDED       51
                       SRB           0  SRB           0.0    HSP MISS    0.00   CENTRAL    0.00   END/SEC   0.06   QUE 000.00.00.000
                       TOT           0  TCB+SRB%      0.0    EXP SNGL    0.00   EXPAND     0.00   #SWAPS       0
                       PER SEC       0                       EXP BLK     0.00                                      TOT 000.00.00.000

SUBSYS = OAM          TRXCLASS =            ACCTINFO = NO
USERID =               TRXNAME = ACCESS
```

Figure 24. Sample RMF Monitor I Workload Activity Report (Part 1 of 4)

```
                                                W O R K L O A D   A C T I V I T Y
                                                                                                                   PAGE    2
            MVS/ESA                     SYSTEM ID 3090            DATE 09/01/93           INTERVAL 12.26.657
            SP5.2.0                     RPT VERSION 5.2.0         TIME 11.15.26           IPS = IEAIPS04
         OPT = IEAOPT00                          SUMMARY BY PERFORMANCE GROUP   SERVICE DEFINITION COEFFICIENTS     SU/SEC=1162.3
         ICS = IEAICSTU                                                         IOC =  5.0  CPU = 10.0  SRB = 10.0  MSO =  3.0000
               TIME                        AVERAGE ABSORPTION,                                                        AVG TRANS. TIME,
PGN  PGP  DMN  SLICE  INTERVAL SERVICE  AVG TRX SERV RATE,     PAGE-IN RATES      STORAGE         TRANSACTIONS        STD. DEVIATION
               GROUP                       TCB+SRB SECONDS, %                                                             HHH.MM.SS.TTT
0401 ALL  ALL  ALL     IOC           0  ABSRPTN         0   SINGLE       0.00  AVERAGE    0.00  AVG        0.00    TRX 000.00.01.039
                       CPU           0  TRX SERV        0   BLOCK        0.00                   MPL        0.00    SD  000.00.00.178
                       MSO           0  TCB           0.0   HSP          0.00  TOTAL      0.00  ENDED        16
                       SRB           0  SRB           0.0   HSP MISS     0.00  CENTRAL    0.00  END/SEC    0.02    QUE 000.00.00.000
                       TOT           0  TCB+SRB%      0.0   EXP SNGL     0.00  EXPAND     0.00  #SWAPS        0
                       PER SEC       0                      EXP BLK      0.00                                      TOT 000.00.00.000
0402 ALL  ALL  ALL     ZEROS

SUBSYS = OAM          TRXCLASS =             ACCTINFO = NO
USERID =               TRXNAME = DELETED
0403 ALL  ALL  ALL     IOC           0  ABSRPTN         0   SINGLE       0.00  AVERAGE    0.00  AVG        0.00    TRX 000.00.00.062
                       CPU           0  TRX SERV        0   BLOCK        0.00                   MPL        0.00    SD  000.00.00.079
                       MSO           0  TCB           0.0   HSP          0.00  TOTAL      0.00  ENDED         5
                       SRB           0  SRB           0.0   HSP MISS     0.00  CENTRAL    0.00  END/SEC    0.00    QUE 000.00.00.000
                       TOT           0  TCB+SRB%      0.0   EXP SNGL     0.00  EXPAND     0.00  #SWAPS        0
                       PER SEC       0                      EXP BLK      0.00                                      TOT 000.00.00.000
0404 ALL  ALL  ALL     ZEROS
THRU
0408
SUBSYS = OAM          TRXCLASS =             ACCTINFO = NO
USERID =               TRXNAME = QUERY
0409 ALL  ALL  ALL     IOC           0  ABSRPTN         0   SINGLE       0.00  AVERAGE    0.00  AVG        0.00    TRX 000.00.00.667
                       CPU           0  TRX SERV        0   BLOCK        0.00                   MPL        0.00    SD  000.00.01.259
                       MSO           0  TCB           0.0   HSP          0.00  TOTAL      0.00  ENDED         5
                       SRB           0  SRB           0.0   HSP MISS     0.00  CENTRAL    0.00  END/SEC    0.00    QUE 000.00.00.000
                       TOT           0  TCB+SRB%      0.0   EXP SNGL     0.00  EXPAND     0.00  #SWAPS        0
                       PER SEC       0                      EXP BLK      0.00                                      TOT 000.00.00.000
SUBSYS = OAM          TRXCLASS =             ACCTINFO = NO
USERID =               TRXNAME = QUERYG
```

Figure 24. Sample RMF Monitor I Workload Activity Report (Part 2 of 4)

```
                                              W O R K L O A D   A C T I V I T Y
                                                                                                                      PAGE    3
          MVS/ESA                     SYSTEM ID 3090           DATE 09/01/93            INTERVAL 12.26.657
          SP5.2.0                     RPT VERSION 5.2.0        TIME 11.15.26            IPS = IEAIPS04
        OPT = IEAOPT00                      SUMMARY BY PERFORMANCE GROUP   SERVICE DEFINITION COEFFICIENTS     SU/SEC=1162.3
        ICS = IEAICSTU                                                     IOC =  5.0  CPU = 10.0  SRB = 10.0  MSO =  3.0000
              TIME                      AVERAGE ABSORPTION,                                                        AVG TRANS. TIME,
PGN  PGP  DMN  SLICE  INTERVAL SERVICE  AVG TRX SERV RATE,     PAGE-IN RATES      STORAGE         TRANSACTIONS      STD. DEVIATION
              GROUP                     TCB+SRB SECONDS, %                                                             HHH.MM.SS.TTT
0410 ALL  ALL  ALL     IOC          0   ABSRPTN        0    SINGLE      0.00   AVERAGE     0.00   AVG       0.00    TRX 000.00.01.305
                       CPU          0   TRX SERV       0    BLOCK       0.00                      MPL       0.00    SD  000.00.00.000
                       MSO          0   TCB          0.0    HSP         0.00   TOTAL       0.00   ENDED        1
                       SRB          0   SRB          0.0    HSP MISS    0.00   CENTRAL     0.00   END/SEC   0.00    QUE 000.00.00.000
                       TOT          0   TCB+SRB%     0.0    EXP SNGL    0.00   EXPAND      0.00   #SWAPS       0
                       PER SEC      0                       EXP BLK     0.00                                        TOT 000.00.00.000
0411 ALL  ALL  ALL     ZEROS
THRU
0412

SUBSYS = OAM          TRXCLASS =           ACCTINFO = NO
USERID =               TRXNAME = RETRVPD
0413 ALL  ALL  ALL     IOC          0   ABSRPTN        0    SINGLE      0.00   AVERAGE     0.00   AVG       0.00    TRX 000.00.00.162
                       CPU          0   TRX SERV       0    BLOCK       0.00                      MPL       0.00    SD  000.00.00.208
                       MSO          0   TCB          0.0    HSP         0.00   TOTAL       0.00   ENDED        2
                       SRB          0   SRB          0.0    HSP MISS    0.00   CENTRAL     0.00   END/SEC   0.00    QUE 000.00.00.000
                       TOT          0   TCB+SRB%     0.0    EXP SNGL    0.00   EXPAND      0.00   #SWAPS       0
                       PER SEC      0                       EXP BLK     0.00                                        TOT 000.00.00.000
0414 ALL  ALL  ALL     ZEROS
THRU
0419

SUBSYS = OAM          TRXCLASS =           ACCTINFO = NO
USERID =               TRXNAME = STORDASD
0420 ALL  ALL  ALL     IOC          0   ABSRPTN        0    SINGLE      0.00   AVERAGE     0.00   AVG       0.00    TRX 000.00.01.735
                       CPU          0   TRX SERV       0    BLOCK       0.00                      MPL       0.00    SD  000.00.03.672
                       MSO          0   TCB          0.0    HSP         0.00   TOTAL       0.00   ENDED        5
                       SRB          0   SRB          0.0    HSP MISS    0.00   CENTRAL     0.00   END/SEC   0.00    QUE 000.00.00.000
                       TOT          0   TCB+SRB%     0.0    EXP SNGL    0.00   EXPAND      0.00   #SWAPS       0
                       PER SEC      0                       EXP BLK     0.00                                        TOT 000.00.00.000
0421 ALL  ALL  ALL     ZEROS
THRU
0422

SUBSYS = OAM          TRXCLASS =           ACCTINFO = NO
USERID =               TRXNAME = UNACCESS
```

Figure 24. Sample RMF Monitor I Workload Activity Report (Part 3 of 4)

```
                                              W O R K L O A D   A C T I V I T Y
                                                                                                                      PAGE    4
          MVS/ESA                     SYSTEM ID 3090           DATE 09/01/93            INTERVAL 12.26.657
          SP5.2.0                     RPT VERSION 5.2.0        TIME 11.15.26            IPS = IEAIPS04
        OPT = IEAOPT00                      SUMMARY BY PERFORMANCE GROUP   SERVICE DEFINITION COEFFICIENTS     SU/SEC=1162.3
        ICS = IEAICSTU                                                     IOC =  5.0  CPU = 10.0  SRB = 10.0  MSO =  3.0000
              TIME                      AVERAGE ABSORPTION,                                                        AVG TRANS. TIME,
PGN  PGP  DMN  SLICE  INTERVAL SERVICE  AVG TRX SERV RATE,     PAGE-IN RATES      STORAGE         TRANSACTIONS      STD. DEVIATION
              GROUP                     TCB+SRB SECONDS, %                                                             HHH.MM.SS.TTT
0423 ALL  ALL  ALL     IOC          0   ABSRPTN        0    SINGLE      0.00   AVERAGE     0.00   AVG       0.00    TRX 000.00.00.051
                       CPU          0   TRX SERV       0    BLOCK       0.00                      MPL       0.00    SD  000.00.00.053
                       MSO          0   TCB          0.0    HSP         0.00   TOTAL       0.00   ENDED       17
                       SRB          0   SRB          0.0    HSP MISS    0.00   CENTRAL     0.00   END/SEC   0.02    QUE 000.00.00.000
                       TOT          0   TCB+SRB%     0.0    EXP SNGL    0.00   EXPAND      0.00   #SWAPS       0
                       PER SEC      0                       EXP BLK     0.00                                        TOT 000.00.00.000
```

Figure 24. Sample RMF Monitor I Workload Activity Report (Part 4 of 4)

RMF Monitor II Transaction Activity Report

The data in the RMF Monitor II Transaction Report is collected by SRM and passed to RMF when an RMF Monitor I session that is measuring workload activity is active. Thus, the transaction activity report can only be obtained when an RMF Monitor I session, requesting workload activity using the WKLD option is running at the same time as the report requested.

Figure 25 shows a sample of the RMF Monitor II Transaction Activity report of OAM transaction completions for various report performance groups.

```
                                  CPU=  3 UIC=183 PFR=   0          TRX      T
11:24:28  ICS=IEAICSTU  IPS=IEAIPS04     ACCT    PERF    TRANS  AVG TRANS TIME
SUBSYS  TRXCLASS   USERID     TRXNAME    INFO   GRP PER   RATE   HHH.MM.SS.TTT
                                              C   2   1   0.079  000.00.00.134
                                                      2   0.004  000.00.00.699
                                                      3   0.041  000.00.05.426
OAM     OSREQ                              NO R 400   1   0.118  000.00.00.616
OAM                           ACCESS       NO R 401   1   0.037  000.00.01.039
OAM                           DELETED      NO R 403   1   0.011  000.00.00.062
OAM                           QUERY        NO R 409   1   0.011  000.00.00.667
OAM                           QUERYG       NO R 410   1   0.002  000.00.01.305
OAM                           RETRVPD      NO R 413   1   0.004  000.00.00.162
OAM                           STORDASD     NO R 420   1   0.011  000.00.01.735
OAM                           UNACCESS     NO R 423   1   0.039  000.00.00.051
```

Figure 25. Sample RMF Monitor II Transaction Activity Report

Activating an Installation Control Specification

SRM must be notified that a new ICS is in effect, after the IEAICS*xx* member is updated with the OAM transaction names that need statistics compiled for them. To notify SRM about the new ICS, the system operator should issue the MVS SET system command:

```
SET ICS=xx
```

The *xx* in the SET ICS=*xx* command refers to the low order digit suffix of the IEAICS*xx* PARMLIB member that contains the SUBSYS specification for the OAM RMF transaction names.

Initializing RMF and Starting a Monitor I Session

To start RMF and a Monitor I session at the same time, the system operator should issue the following command at any MVS system console:

```
START RMF.A
```

RMF must be collecting workload activity statistics, so the RMF WKLD options must be specified either:

- On the START RMF command issued at an MVS system console, **or**
- In the RMF PARMLIB member ERBRMF*xx* that is processed by RMF during initialization. If not specified, the default is ERBRMF00.

Note: Remember the WKLD(PERIOD) RMF option must be in effect requesting reporting by performance group period or the WKLD(GROUP) RMF option must be in effect requesting reporting by performance group on the Workload Activity Report.

Starting and Ending an RMF Monitor II Session

To start an RMF Monitor II session:

- Log on to TSO.

- Enter `RMFMON`.

To end the RMF Monitor II Session:
- Enter the stop command `Z`.

You can then log off of TSO or continue to do other TSO work.

Obtaining an RMF Monitor II Transaction Activity Report

To obtain the RMF Monitor II Transaction Activity Report:
- Press the PF12 key to select the Transaction Activity Data menu item from the RMF DISPLAY MENU, **or**
- Enter the command `TRX OAM` on the RMF DISPLAY MENU to display the transaction activity report containing statistics for all transactions associated with the subsystem name OAM. For complete information about the TRX command and other RMF Monitor II menu items and display commands, refer to *RMF User's Guide*.

Note: On the first invocation of the SYSEVENT macro following an IPL, message CBR7012 is returned indicating a return code of "8". This should be expected as it indicates that SRM has not yet acquired data storage buffers for recording transaction completion messages. The initial failing request is not reported to RMF.

Establishing Recovery Procedures

As part of your disaster recovery plan, establish and test procedures for recovering these entities:
- DB2 object storage databases
- The optical configuration database
- Single objects
- An entire optical or tape volume
- Accessing backup objects automatically
- Collection name entries in a catalog

Recovering DB2 Databases

The recoverable structure of data in DB2 is the table space. To ensure recoverability, make an image copy when creating each table space in the optical configuration database and all the table spaces in each of the object storage databases. For directions regarding how to make these image copies, refer to *DB2 for OS/390 Administration Guide*. Your installation determines how often to make backup copies, based on the usage of each table space. Use this original image copy as a base, and make subsequent periodic incremental image copies of each table space.

At specified intervals (best defined based on the usage of each table space), perform a MERGECOPY on the base (original, full-image copy) and subsequent incremental-image copies to establish a new base. After creating the new base level, perform subsequent incremental image copies in relation to this new base.

The main benefit of periodically using MERGECOPY to create a new base is the time savings at the time of the failure. Merge copies can be time-consuming, so it is best to do them on a timely, convenient basis.

To recover a table space, merge the contents of the DB2 recovery log with the most recent full-image copy of the table space. Because each change made to the database is recorded in the DB2 recovery log, the merge restores the table space to its last point of consistency prior to system failure.

Notes:

1. In DB2, *point of consistency* is a term that designates a time when all recoverable data accessed by an application program is consistent with other data. It is also known as *syncpoint* or *commit point*. Refer to *DB2 for OS/390 Administration Guide* for more information.
2. The entries within the DB2 collection name table are synchronized with a corresponding collection name entry in the catalog. Therefore, recovery of the DB2 collection name table must result in a table consistent with the catalog.
3. If any action is taken that permanently removes an entry from the DB2 collection name table, the corresponding entry must be deleted from the catalog. After a collection entry is removed from the DB2 collection name table and the catalog, objects contained within the collection are no longer accessible or managed by OSMC.

Recovering Single Objects from Removable Media

OAM contains a single object recovery utility for recovering a single object from removable media. You can use either the first or the second backup copy for single object recovery as determined by the settings of the SETOSMC statements in the currently active CBROAM*xx* member of PARMLIB. The system creates a new primary copy from a backup copy (if one exists) using the following criteria:

- If the primary object resides on optical disk, OAM uses a backup copy (on either optical disk or tape) to create a new optical primary copy.
- If the primary object resides on tape, a backup copy (on either optical disk or tape) is used to create a new tape primary copy.
- If the primary object resides on DASD, OAM uses a backup copy (on either optical disk or tape) to create a new DASD primary copy.

The operator starts the single object recovery utility to copy the object. For further information on this procedure, see “Starting Object Recovery for Single Objects” on page 235.

Recovering an Entire Optical Cartridge or Tape Volume

OAM contains a utility program that recovers the objects from an unusable optical or tape volume to a usable volume. This utility is called Volume Recovery utility. The Volume Recovery utility is used in the event that an optical or tape volume is rendered unreadable, either because of physical damage, or because the volume cannot be found. The Volume Recovery utility is used for two types of volume recovery:

- Volumes containing primary objects belonging to an Object storage group can be recovered from the first or second backup copies of the objects (optical or tape).
- Backup volumes belonging to an Object Backup storage group can be recovered from the primary copies of the objects (DASD, optical, or tape). All storage groups that contain objects that need to be recovered must be defined as part of the ACDS configuration.

To recover a primary optical or tape volume, all of the backup volumes containing either the first or the second backup copies of the objects on the primary volume are needed, whether they are optical or tape. The media from which the backup copy is retrieved for the recovery depends on the volume on which the selected backup copy resides.

For example, OAM may have run one storage management cycle for the storage group after OAM was initialized with a CBROAM*xx* PARMLIB member that contained a particular SETOAM statement. This SETOAM statement specified a tape unit name for an Object Backup storage group that caused OAM to write backups for that group to tape. Another time, OAM may have run a storage management cycle for the storage group after OAM was initialized, and the system invoked the START OAM command with one of these three options:

- Without a CBROAM*xx* PARMLIB member
- With a CBROAM*xx* PARMLIB member that contained no SETOAM statements
- With a SETOAM statement that did not specify a tape unit name that was associated with an Object Backup storage group

Any one of these options will cause OAM to write backups to an optical volume.

When recovering a backup volume, every Object storage group must be searched for primary objects having backup copies residing on the backup volume being recovered. The primary copy for each of these objects can be on DASD, optical, or tape. As a result, the Volume Recovery utility must identify the optical volumes as well as the tape volumes that are needed for recovery. If both optical and tape volume are requested for the recovery, the operator must reply that both types are available for the recovery to continue.

The operator starts the recovery utility to copy the data. For further information on this procedure, see “Starting the OAM Volume Recovery Utility” on page 229.

Accessing Backup Objects Automatically

OAM allows your application to automatically obtain the first or second backup copy of an object when the primary copy of the object resides on one of these media:

- On removable media that is marked not readable (possibly damaged or destroyed)
- On removable media that is in a library that is offline or pending offline
- On removable media that is in a library that is not operational

When you activate this function for one or more of the above conditions, and that condition exists when the OSREQ application interface retrieves an object, OAM attempts to obtain the first or second backup copy, if one exists. The backup copy that is selected for retrieval is determined by the BACKUP1 **|** BACKUP2 keyword that is specified on the MODIFY OAM,START,AB command. If this automatic access to backup objects function is inactive and the primary copy of the object is not available for any of the above reasons, the OSREQ API passes error return and reason codes back to the application. If no backup copy exists and the function is active, the OSREQ API passes error return and reason codes back to the application. Automatic access to backup does not automatically force all retrieves to be initiated from backup.

The operator activates and deactivates this function through an operator command. For further information on this function, see “Starting Automatic Access to Backup Copies of Objects” on page 241.

Recovering Collection Name Catalog Entries

OAM attempts to keep collection name catalog entries up to date. This cannot be accomplished if the catalog entry does not exist or if the catalog is unusable (for example, because of I/O errors). Recovery of the catalog may be required. Standard catalog recovery procedures apply to recovering catalog entries for collection names. Those procedures usually involve making an image copy (for example, IDCAMS EXPORT) at certain intervals and restoring that copy (for example, IDCAMS IMPORT) to recover an unusable catalog.

If collection name catalog entries were added after the image copy was made, restoring an image copy does not complete the recovery; you must also recreate those added entries. When a collection name entry from the collection name table is lost, objects in that collection will not be processed in a storage management cycle. If you do not have a program or program product to apply a journal of those additions, you can use IDCAMS to recatalog those individual entries.

Note: For further information on the use of IDCAMS with collection name entries in the catalog, refer to the *z/OS DFSMS Access Method Services* manual.

Using the Move Volume Utility

OAM provides a utility, the Move Volume utility (MOVEVOL), capable of moving objects from a primary or backup source volume (a tape volume or one side of an optical disk) to one or more target volumes. The set of eligible target volumes is determined by the drives eligible to write data into the Object or Object Backup storage group containing the source volume. The set of drives eligible to write data into the storage group containing the source volume is defined by the storage management subsystem (SMS) storage group construct definition and the corresponding SETOAM statement if it exists.

If the source volume is an optical volume belonging to an Object storage group, the target volumes are optical volumes belonging to the same storage group. If there are no optical drives available to write to the Object storage group to which the optical source volume belongs, the MOVEVOL request fails.

If the source volume is a tape volume belonging to an Object storage group, the target volumes are tape volumes belonging to the same storage group. If there is not a valid tape unit name associated with the Object storage group to which the tape source volume belongs, the MOVEVOL request fails.

If the source volume is either an optical or a tape volume belonging to the Object Backup storage group, the type of media used for writing the objects is derived from the definition of the Object Backup storage group. For example, if there is a valid SETOAM statement for the Object Backup storage group to which the source volume belongs, the target volumes chosen are tape volumes. If there is no valid SETOAM statement for the Object Backup storage group belonging to the source volume, the target volumes chosen are optical disk volumes.

The Move Volume utility performs the following functions:
- Writes objects using the optical drives defined in the Object storage group when moving objects off a primary optical source volume
- Writes objects using the tape drives allocated to the Object storage group (using the tape unit name specified in the SETOAM statement for the Object storage group) when moving objects off a primary tape source volume
- Writes objects using optical drives defined to, or tape drives allocated to the Object Backup storage group when moving objects off of a backup optical or tape source volume

For media migration purposes of objects in the Object storage group the Move Volume utility enables you to move data from the following:
- One optical media type to another optical media type
- One tape media type to another tape media type

For media migration purposes of objects in the Object Backup storage group the Move Volume utility enables you to move data from the following:

- One optical media type to another optical media type
- One tape media type to another tape media type
- Optical media to tape media
- Tape media to optical media

The intent of the utility is to facilitate migration from “older” technology media to “newer” technology media (for example, from 12-inch media to 5.25-inch media). However, since the set of drives eligible to write data is defined by the definition of the storage group, the following examples of data movement in Table 32 are also supported.

Table 32. Examples of Data Movement with the Move Volume Utility

SCENARIO	MOVE FROM	MOVE TO
Movement from one optical media type to another optical media type	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8x-density WORM 8x-density REWR	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8x-density WORM 8x-density REWR
Movement from one tape media type to another tape media type	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance
Movement from any optical media type to a tape volume belonging to the Object Backup storage group	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8x-density WORM 8x-density REWR	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance
Movement from any tape media type to optical media type belonging to the Object Backup storage group	Cartridge System Tape Enhanced Capacity High Performance Extended High Performance	single-density WORM single-density REWR double-density WORM double-density REWR quad-density WORM quad-density REWR 8x-density WORM 8x-density REWR
Note: WORM = write-once-read-many		

Once the data is successfully moved from the source volume, the data is no longer accessible from the original source volume.

Note: If the intent is to migrate from one tape media type to another tape media type (for example, from IBM Cartridge System Tape to IBM Enhanced Capacity Cartridge System Tape), it may be necessary to modify tape volume records in the DB2 TAPEVOL table (see “Updating Fields in the DB2 Volume Table and the Tape Volume Table” on page 315). If you change the

TAPEUNITNAME on the SETOAM statement that assigns the storage group, OAM recognizes the change; however, OAM only uses it for new scratch allocations. If there are available usable tape volumes that belong to this storage group (the storage group to which the source volume belongs), OAM continues to use these volumes until they are all full. In order to force writes to go to a new media type (honoring the changed TAPEUNITNAME), any available usable volumes in that storage group must be either marked not writable or marked full with their percent full set to 100. This causes OAM to believe that the storage group is out of space and to request a scratch allocation using the new SETOAM TAPEUNITNAME specified for that group.

Preparation of the Move Volume Utility Environment

Prior to invoking the Move Volume utility, do the following to prepare the environment in which the utility is executed:

- Update and activate a source control data set (SCDS), if necessary.
- Minimize or quiesce contending system activity.
- Analyze resources and tune OAM.
- Ensure that there is a valid SETOAM statement for the Object Backup storage group if you plan to use the Move Volume utility to move objects from the Object Backup storage group to a tape volume.
- Ensure that there is an optical library and drives defined to the Object Backup storage group if you plan to use the Move Volume utility to move objects from the Object Backup storage group to an optical volume.
- Ensure that there is a valid SETOAM statement for the Object storage group to which the source volume belongs if the source volume is a tape volume.
- Ensure that there is an optical library and drives defined to the Object storage group to which the source volume belongs if the source volume is an optical volume.

Each of these steps are explained in further detail below.

Updating and Activating an SCDS

In your preparation for invocation of the Move Volume utility, update the SCDS, if necessary, for the following conditions that apply to your usage of the utility and then activate the updated SCDS:

- If the source volume is a backup volume from the Object Backup storage group, then the SCDS must include *all* Object storage groups containing objects that have a backup copy on the backup source volume. The utility uses the list of Object storage groups defined in the SCDS to identify the objects to be moved.

  Failure to include all Object storage groups containing objects that have a backup copy on the backup source volume causes invalid results. For example, although all of the objects have not been moved from the source volume (due to exclusion in the SCDS of some of the Object storage groups containing objects having a backup copy on the backup source volume), the utility may issue messages indicating that all objects are moved from the source volume because all the objects in the Object storage groups identified in the SCDS *have been* moved.
- The storage group containing the source volume may need to be updated to define the set of drives eligible to write data into the storage group. Make sure that the storage group containing the source volume includes libraries capable of writing to the target optical media type. You can accomplish this through one of the following methods:

- Remove libraries from the storage group definition that contains volumes that you do not want to be used as target volumes for MOVEVOL. Add libraries that contain volumes that you do want to be used as target volumes for MOVEVOL.
- Leave the libraries with the undesired media assigned to the storage group, but for the duration of MOVEVOL, mark them as unwritable or full.
- With only multifunction device types in the storage group definition, exploit the default media type in the library definition to direct writes to a specific media type.
- With only multifunction device types in the storage group definition, exploit the library's read-only and write capable function based on the source and target media types.

The following examples (Table 33, Table 34, Table 35 on page 198, and Table 36 on page 198) show how to migrate data from an older media type (OMT) to a newer media type (NMT) using various methods:

Table 33. Migrating from Old Media Type to New Media Type by Removing Libraries from Storage Group Definitions

SCENARIO	You have libraries with drives capable of writing to OMT defined to a storage group.
DESIRED AFFECT	To ensure that the target volumes for the MOVEVOL request are NMT.
TASK	• Remove any libraries with drives capable of writing to OMT from the storage group definition. • Add libraries with drives capable of writing to NMT to the storage group definition, if there aren't some currently defined.

Table 34. Migrating from Old Media Type to New Media Type without SCDS Modifications

SCENARIO	VOL1 is OMT and belongs to SG1. Libraries containing drives capable of writing to both OMT and NMT exist in the SCDS.
DESIRED AFFECT	To ensure that the target volumes for the MOVEVOL request for VOL1 are NMT.
TASK	• Mark full or unwritable any scratch volume in libraries associated with SG1 that are capable of writing to OMT. • Mark full or unwritable any volume belonging to SG1 in libraries that are capable of writing to OMT.

Table 35. Migrating from Old Media Type to New Media Type using Library Default Media Type

SCENARIO	All libraries associated with the source volume's storage group are multifunction 3995 optical libraries that are capable of writing to both OMT and NMT.
DESIRED AFFECT	To ensure that the target volumes for the MOVEVOL request are NMT.
TASK	Modify the library default media type to a media type that includes NMT but not OMT for each library associated with the source volume's storage group. This allows read requests from OMT to still occur inside the libraries, however, any new requests to that storage group will be satisfied with NMT.

Table 36. Migrating from Old Media Type to New Media Type using Multifunction Libraries and no SCDS Modification

SCENARIO	You are using multifunction 3995 optical libraries that support OMT as read-only.
DESIRED AFFECT	To ensure that the target volumes for the MOVEVOL request are NMT.
TASK	Make certain that no libraries associated with the source volume's storage group support writing to OMT. The reads from the source volume are still supported and all writes are directed to NMT without making any modification to the SCDS.

For more information regarding updating, validating, and activating an SCDS, refer to *z/OS DFSMSdfp Storage Administration Reference*.

Minimizing or Quiescing Contending System Activity

Prior to and during the execution of the Move Volume utility, you should minimize or quiesce the following types of system activity, which may interfere with the utility:

- Any activity involving *reading* from the source volume. Or, if the source volume is an optical volume any activity involving *reading* from the volume on the opposite side of the optical disk. If read activity on the source volume cannot be quiesced, investigate using the UPD=N option on the CBRINIT statement in the IEFSSN*xx* member of PARMLIB during initial program load (IPL). This will avoid conflicting updates to the objects' directory entries.
- Any activity involving *inserting, selecting, updating, or deleting* operations on the OAM DB2 object directory table for the storage group

System activity includes, but is not limited to the following:

- Scheduled or operator-initiated OAM Storage Management Component (OSMC) storage group management cycles involving the:
  - Storage group containing the source volume when moving objects from a primary volume
  - Storage groups containing objects that have a backup copy on the source volume when moving objects from a backup optical volume
- Operator-initiated invocations of the Move Volume utility for other volumes requiring the same OAM DB2 object directory tables

- Operator-initiated invocations of the Volume Recovery utility for other volumes requiring the same OAM DB2 object directory tables
- Applications using the OSREQ interface to read data from or write data to the storage groups referenced by the Move Volume utility, or both

Failure to minimize or quiesce this type of system activity may cause operations in the other system activities to complete unsuccessfully (for example, due to the unavailability of the source volume during the execution of the utility) or may interfere with the operation of the utility (for example, causing contention in the DB2 databases required for the utility to operate).

Analyzing Resources and Tuning OAM for MOVEVOL Usage

At a minimum, planning should be performed to use the Move Volume utility when contending system activity is low.

When planning to use the Move Volume utility, the same considerations used for planning the number of OSMC storage group management cycles to run concurrently apply. The Move Volume utility is a long-running process that, for planning purposes, requires a drive for reading and at least one drive for writing for its duration. You may have to tune your existing implementation of OAM for these considerations using the MAXS and or DRIVE STARTUP threshold, or both controls as described in “MAXS Considerations for MOVEVOL Processing” and “DRIVE STARTUP Threshold Considerations for MOVEVOL Processing”. For more information on tuning OAM, see “Tuning OAM” on page 170.

MAXS Considerations for MOVEVOL Processing

Review your specification for MAXS. The Move Volume utility is not controlled by the MAXS value; however, it is a consumer of resources. It is not recommended that you run the Move Volume utility at the same time as an OSMC storage management cycle since this causes a conflict for resources. However, if you plan to run the Move Volume utility at the same time as one or more OSMC storage management cycles, you may need to reduce the number of concurrent OSMC storage management cycles to make resources available for the Move Volume utility. Consider and account for the resources that the Move Volume utility requires first and then distribute the remaining resources for the OSMC storage management cycle with MAXS.

Also, because the MAXS value does not control the Move Volume utility, take resource considerations into account when planning for concurrent executions of the utility (for example, multiple Move Volume utilities running concurrently for different volumes).

DRIVE STARTUP Threshold Considerations for MOVEVOL Processing

Always review your specifications for DRIVE STARTUP threshold and TAPEDRIVESTARTUP for your Object storage groups. The Move Volume utility causes objects to be written to the storage group of which the source volume is a member.

For example, if the Move Volume utility is running at the same time as one or more OSMC storage management cycles for the same storage group, the write activity in the storage group is increased. As a result, you must consider and account for the drives that the Move Volume utility requires in addition to the drives that are required by the OSMC storage management cycle when determining the
| appropriate value for the threshold. You must perform this action for each Object
| storage group containing primary source volumes that are processed by the Move

Volume utility as well as for any Object Backup storage group for backup source volumes that are processed by the Move Volume utility.

Attention: Running the Move Volume utility concurrently with the OSMC processing for the storage group that the volume belongs to can cause DB2 contention.

DB2 Index Considerations

There are several factors that influence the performance of the Move Volume utility. The following are items to consider when making processing decisions to maximize the performance of the utility:

- Average object size
- Source and target optical device types
- Number of active utilities
- Primary or backup source volume
- Number of rows in the object directory table for the storage group to which the source volume belongs
- Additional DB2 indexes defined (depending on activity currently in progress)
- Level of the system workload and I/O to the source and target devices

The Move Volume utility performs DB2 SELECTs from the object directory tables (OSM_OBJ_DIR) to prepare the list of objects to be moved. Depending on the current status of the table, DB2 may do a table space scan or an index scan. With large DB2 tables, it is recommended that you verify that DB2 will be using index scans to enhance the efficiency and performance of the Move Volume utility.

The object selection is not overlapped with the read from optical or tape, write to optical or tape, or directory update processes. This means that the elapsed time to complete the MOVEVOL processing consists of the time to select the objects and the time to update the object directory tables after the object movement is complete.

The object selection process consists of executing DB2 SELECT statements from the object directory tables. This process is limited by the DB2 indexes available and the chosen access path. Without the additional recommended indexes, the object selection process requires:

- at least one table space scan of the object directory table for a single storage group associated with the primary volume processed
- a table space scan of *all* object directory tables when a backup volume is processed.

Note: Creating these additional DB2 indexes will enhance the performance for the Move Volume utility; however, these indexes can negatively affect other OSMC process. After MOVEVOL processing has completed, drop these additional indexes and rebind the modified plans.

To improve performance for the Move Volume utility, perform the following recommended steps:

1. Create an index on ODLSLOC and ODCLID for each storage group that contains a primary volume that is being processed. For first backup volumes, create an index on ODBKLOC and ODCLID for each storage group that has a significant number of objects (more than 10 000). For second backup volumes, create an index on ODBK2LOC and ODCLID for each storage group that has a

significant number of objects (more than 10 000). Use the following SQL statements to define additional DB2 indexes. See “CBRISQL0” on page 390 for sample job control language (JCL).

```
  CREATE INDEX GROUPhlq.OBJDIRX4
         ON    GROUPhlq.OSM_OBJ_DIR
         (
          ODLSLOC   ASC,
          ODCLID    ASC
         )
         USING      VCAT cat_name
         CLOSE      NO
         SUBPAGES   1
        BUFFERPOOL BP1
        PCTFREE    10;
 CREATE INDEX GROUPhlq.OBJDIRX5
        ON    GROUPhlq.OSM_OBJ_DIR
        (
         ODBKLOC   ASC,
         ODCLID    ASC
        )
        USING      VCAT cat_name
        CLOSE      NO
        SUBPAGES   1
        BUFFERPOOL BP1
        PCTFREE    10;
```

where: *cat_name* is the name of the catalog used under DB2. OAM does not create these indexes using the UNIQUE keyword.

Because moving data from a backup volume requires access to all of the defined storage group directories, creation of the index on ODBKLOC (instead of ODLSLOC as in above example) is important for all storage groups that contain large numbers of objects (whether there are any backup objects on the volume being processed or not).

2. Execute RUNSTATS to collect data on the new indexes. See “Tuning the DB2 Databases” on page 170 for more information.
3. Rebind the following DB2 plans: CBRHSVOL, CBRHSBKV, CBRHSBCC, CBRHSPCC. See “CBRHBIND and CBRHGRNT” on page 433 for information on binding these plans.
4. Define data sets for the additional indexes. See “CBRIALC0” on page 383 for more information on defining these data sets.

```
//STEPxx  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSDUMP  DD SYSOUT=*
//SYSIN    DD *
 DELETE
 cat_name.DSNDBC.GROUPhlq.OBJDIRX4.I0001.A001
     CLUSTER
     PURGE
 DELETE
 cat_name.DSNDBC.GROUPhlq.OBJDIRX5.I0001.A001
     CLUSTER
     PURGE
 SET LASTCC=0
 SET MAXCC=0
 DEFINE CLUSTER
 (NAME(cat_name.DSNDBC.GROUPhlq.OBJDIRX4.I0001.A001)
          LINEAR
          SHAREOPTIONS(3 3)
          VOLUMES(vol_ser)
          CYLINDERS(pri_alloc sec_alloc)
          UNIQUE   )
        DATA
 (NAME(cat_name.DSNDBD.GROUPhlq.OBJDIRX4.I0001.A001))
 DEFINE CLUSTER
 (NAME(cat_name.DSNDBC.GROUPhlq.OBJDIRX5.I0001.A001))
         LINEAR
         SHAREOPTIONS(3 3)
         VOLUMES(vol_ser)
         CYLINDERS(pri_alloc sec_alloc)
         UNIQUE    )
       DATA
 (NAME(cat_name.DSNDBD.GROUPhlq.OBJDIRX5.I0001.A001))
/*
```

where: *vol_ser* is the volume serial the data set should be placed on; *pri_alloc*, *sec_alloc* is the number of primary and secondary allocations for the data set; *cat_name* is the name of the catalog used under DB2. These indexes are not created using the UNIQUE keyword.

5. Run the Move Volume utility.
6. Drop the DB2 indexes used during MOVEVOL processing.
7. Execute RUNSTATS again.
8. Rebind the following DB2 plans: CBRHSVOL, CBRHSBKV, CBRHSBCC, CBRHSPCC.

Migrating objects from one media to another takes a significant amount of time. With this in mind, there are some things that you should consider that could save time and effort in the future.

- Define expiration criteria for all objects. This should reduce the amount of data and time needed to perform migration to another media.
- Adopt higher capacity media into your installation as soon as it is introduced. If you write all objects to the highest capacity media available, you may be able to avoid a media migration.

Processing Object Expiration

Each object stored by OAM is assigned an expiration date. This expiration date is derived using the retention period (RETPD) keyword on the OSREQ STORE macro when the object was stored, the expiration rules in the SMS management class assigned to the object, or both.

When a class transition occurs, the SMS storage class and management class ACS routines are invoked. The SMS storage class ACS routine can assign a new management class to the object. Input to the SMS storage class and management class ACS routine indicates that the reason the routine is invoked is for an OAM object class transition. As a result of a new SMS storage class being assigned to the object, the physical location of the object in the OAM storage hierarchy may change. As a result of a new SMS management class assignment to the object, the expiration date of the object may change as well as its backup requirements.

If an OSREQ CHANGE request is performed and a new retention period is specified with the RETPD keyword on the OSREQ CHANGE macro, then the expiration date of the object is recalculated based on the period specified with the RETPD keyword on the OSREQ CHANGE macro. This is true regardless of the media type of the primary or backup copy.

When the expiration date of an object is reached, OAM invokes the Auto-Delete Installation Exit to approve the deletion of the object. If the exit approves the deletion of the object, the object is expired. When an object expires, the row for the object in the Object Directory Table is deleted and any reusable resources are reclaimed. For objects residing on tape volumes, the number of logical KB deleted from the tape volume is incremented for each object deleted.

When the OSMC storage management cycle determines that an object should be expired, all copies of the object are deleted.

If at any time an object's management class results in the object's expiration date being set to 9999/12/31 while that object is on removable media, that volume's expiration date will be set to 9999/12/31. This will cause the volume to never expire, even if the object's management class changes at a later date allowing the object to expire. Be aware of the affects of expiration dates that can be set by a management class, even if it is being used as an interim management class for an object.

For primary copies on DASD, OSMC deletes the copy from DASD. For primary copies on optical and tape, OSMC makes a delete request to LCS to delete the copy from optical or tape.

| For backup copies on optical or tape, OSMC makes a delete request to delete all
| the copies from optical or tape.

For both optical and object tape support, the number of logical KB of data that is deleted from an OAM optical or tape volume containing objects is calculated and stored in the VOLUME or TAPEVOL table. Each time LCS receives a request to delete an object from an optical or tape volume, LCS updates the number of logical KB deleted for that volume. Because an application could choose to do a DB2 ROLLBACK after requesting a delete, the count of the logical KB deleted in the VOLUME or TAPEVOL table is an approximation.

Destroying and Deleting Expired Data

Deleting an optical-disk-resident object from WORM media removes the directory entry for that object; however, the data itself remains on the disk, because of the write-once characteristic of the media. To ensure that confidentiality is maintained, it may be necessary to physically destroy disks that contain expired sensitive material. Follow the manufacturer's directions to safely dispose of optical media.

When a delete request is issued for an object that resides on rewritable optical media, and the directory table entry for the object is deleted upon approval of the Auto-Delete Installation Exit, a row is added to the Deleted Object table in DB2 to indicate the object is to be deleted (physically erased). If the volume where the object resides is selected for a write request, LCS deletes all objects on the volume indicated within the Deleted Object table. LCS also deletes groups of objects when a drive completes a request and there are no other higher priority requests to perform.

OAM does not support expiration of tape volumes; however, a volume expiration date is calculated for tape volumes. It is the responsibility of the installation to ensure that the tape management system does not expire OAM tapes. OAM expects previously used OAM tapes to continue to be available for its use. One possible way to ensure this is to use the expiration date or retention period options on the DATACLASS parameter of the SETOAM statement used with the Object or Object Backup storage group to which the tape volume belongs. Error messages are issued if the control block for a tape volume known to OAM cannot be located, or an error occurs when updating the expiration date for a known OAM tape volume.

NOT Programming Interface information

When OAM processes physical deletes of objects as background work in periods of inactivity, the order of volume preference is as follows:
- Deletions from the mounted volume
- Deletions from the opposite side of the mounted volume
- Deletions from the volume which has the largest amount of space to be deleted (valid only when considering library resident drives)

End of NOT Programming Interface information

Diagnosing Nondeletion/Expiration of Objects During OSMC

This information provides suggestions for handling the situation of objects not deleting or expiring during the OSMC processing cycle. This information allows you to identify, investigate, and overcome these problems.

Objects Not Deleting During OSMC Processing

If, during OSMC processing, objects that should have expired do not actually get deleted from the database, one of two situations is probably causing the problem:
- The auto-delete installation exit (CBRHADUX) has been invoked and completed with a return code not equal to or greater than 8.
- The object has not been selected for expiration processing by OSMC.

To diagnosis the problem, examine the system console log from the start of the OSMC processing of the storage group containing the object through the completion of the OSMC storage group processing. Look for any CBR*xxxxx* messages that may indicate a problem. The CBR*xxxxx* messages are documented in *z/OS MVS System Messages, Vol 4*.

Checking CBRHADUX

If you have not successfully deleted objects during OSMC in the past, or you have just moved to a new release of OAM, then examination of Auto-Delete Installation Exit, CBRHADUX, is a good place to start. For more information on this installation exit, see “Auto-Delete Installation Exit (CBRHADUX)” on page 509. In order to allow objects to be deleted during OSMC you must:

- Modify CBRHADUX to allow deletes.

  Note: In DFP 3.3.0 and later, CBRHADUX must be modified to allow deletes. CBRHADUX returns with a return code 12 if the exit cannot dynamically allocate the group X.OBJECT.DELETE.NOTIFY data set. Check the syslog for CBR, IGD or IEF messages following the start of OSMC processing for the storage group. For additional information on modifying CBRHADUX, see “Auto-Delete Installation Exit (CBRHADUX)” on page 509.
- Refresh LINKLIB.
- Stop and start OAM so that the new copy of CBRHADUX is available to OAM.

Objects Not Selected for Expiration Processing by OSMC

There may be a number of reasons why an object is not selected for expiration during OSMC processing. The expiration date assigned to an object may not be correct, the object may be in a management class that may have different expiration attributes assigned to it than you think it has, a collection entry may not be found within the collection table, or there may be inconsistencies in a collection ID between the catalog and the collection table. The first step in investigating why OSMC does not select the object for expiration processing is to examine the object directory table (HLQ.OSM_OBJ_DIR). The following SQL command retrieves the object’s directory table entry:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE ODNAME = 'object_name';
```

Verify that the pending action date (ODPENDDT) is the current date or earlier and verify that the object has expired.

If the expiration date (ODEXPDT) in the object directory table is less than or equal to the current date, then the object should expire. If the expiration date is the special value 9999-12-31, the object never expires. If the expiration date is 0001-01-01, OAM uses the management class attributes to determine the expiration date.

During OSMC processing, a management cycle called the shelf manager runs after storage management has completed for each Object storage group. Shelf manager examines all the volumes in an Object storage group to determine if there are any in which all objects are expired. If so, the volume is scheduled for ejection and the operator is prompted to remove it from the input/output station of the optical library.

This process does not occur for the Object Backup storage group as OSMC does not run storage management against this storage group. To eject a volume belonging to the Object Backup storage group containing expired objects on both sides of the volume, you must issue an SQL statement for *every* Object storage group in the object directory table to determine if any volumes exist that contain active objects of the expired backup copies. If an active object exists in any Object storage group, the volume for the Object Backup storage group cannot be deleted from the Volume table. If the SQL search of the object directory table against each Object storage group indicates that there are no volumes containing active objects for the backup objects on the Object Backup storage group volume (both sides), the volume can be manually ejected from the library by the operator and the cartridge can be deleted from the Volume table. You must delete both the VOLSER and OVOLSER rows from the Volume table using SQL.

Management Class and Expiration Attribute Definitions

To determine the management class name of the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_MGT_CLASS_TBL M, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODMCNUM = M.ODMCNUM;
```

List the ISMF definition of the management class. Be sure to specify ‘ACTIVE’ as the SCDS name. Examine the expiration attributes. If both EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS are NOLIMIT, the object never expires. The value of EXPIRE AFTER DAYS NON-USAGE is added to the last referenced date (ODLREFDT) in the object directory table to calculate the expiration date. If the last referenced date has not been set, the creation date is used. If the value of EXPIRE AFTER DATE/DAYS is an explicit date, that date is used as the expiration date. If the value is a number of days, the expiration date is calculated based on the number of days added to the creation date (ODCREATS) in the object directory table. For additional information on expiration attribute processing, refer to *z/OS DFSMSdfp Storage Administration Reference*.

Collection Entry Not Found in the Collection Table

Another common cause of an object not being selected by OSMC is the collection entry not being found in the collection table (OAMADMIN.OBJ_COLLECTION_TBL) for the collection ID (ODCLID) in the object directory table for the storage group being processed.

You can verify that the collection table entry exists for the object with the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_COLLECTION_TBL C, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODCLID = C.ODCLID;
```

One row should be returned identifying the collection that you expect to be associated with the object.

Inconsistencies between the Catalog and the Collection Table

Another possibility is a mismatch between the collection ID in the catalog and the collection ID in the collection table. OAM checks for this condition when the OAM address space is started. Check your syslog during OAM address space initialization for message CBR9030I, or any other messages that might indicate a problem with the catalog.

The following TSO command lists the collection entry in the catalog:

```
LISTCAT ENTRIES('collection_name') ALL
```

Verify that the DIRECTORYTOKEN matches the Object storage group name for qualifier HLQ on the ISMF storage group definition panel. Be sure to use `ACTIVE` as the SCDS name when listing Object storage group information. The Object storage group name must also match storage group name (ODCLSGNM) in the collection table.

Documentation for Your IBM Representative

If you have not identified the reason for objects not being deleted during OSMC, you may wish to contact the IBM support center. Make sure that you have the following documentation available:

1. The contents of the following tables:
   - OAMADMIN.CBR_MGT_CLASS_TBL
   - OAMADMIN.CBR_STO_CLASS_TBL
   - OAMADMIN.CBR_COLLECTION_TBL
2. The contents of an object directory table entry for one of the objects that you expected would be deleted.
3. The collection associated with the object directory table entry printed in step 2. The following command provides this information to you:

   ```
   LISTCAT ENTRIES('collection_name') ALL
   ```

4. A screen print of the storage class and management class definitions using `ACTIVE` as the SCDS name.
5. The management class and storage class ACS routines.
6. The syslog from the time the OSMC message CBR9200I is issued indicating the start of processing for the Object storage group until message CBR9201I is issued indicating processing has completed.

Diagnosing Unexpected Results of Object Movement during OSMC Processing

During the course of OSMC processing, objects may not always transition within the storage hierarchy as expected, in spite of the performance objectives of the assigned storage class and management class for the object. Should an installation experience differences between its expectations and the actual location of the object within the hierarchy after OSMC processing, the following information may assist in diagnosing why objects are not moved as expected.

Objects Not Moved to a New Storage Level During OSMC Processing

This information allows you to identify, investigate, and overcome the problem when objects do not move to the expected level within the hierarchy as defined by the ISMF storage class and management class definitions.

To determine where an object resides within the storage management hierarchy (on DASD, optical, or tape), query the directory table entry for the object using the following SQL statement:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE ODNAME = 'object_name';
```

The content of the ODLOCFL field determines the location of the primary copy of the object as follows:

- D—primary copy resides on DASD
- T—primary copy resides on a tape volume
- blank—primary copy resides on an optical disk volume

If the object is not found on the expected medium, it may be because an error has occurred during the OSMC cycle or because the SMS environment has not been properly defined to allow objects to make a transition between storage classes.

To determine if there has been an error during the OSMC cycle, examine the system console log from the start and through the completion of OSMC processing of the Object storage group containing the object.

Any CBR*xxxx* messages found in this log may indicate a problem. For more information concerning these messages and how to resolve the problems that prompted these messages, refer to *z/OS MVS System Messages, Vol 4*.

Objects Not Selected for Class Transition Processing

If once the system console log is examined no error messages are found, the possibility of OSMC not selecting the object for processing needs to be investigated.

The first step in determining if OSMC did not perform a class transition against the object, and why, is to examine the object's directory table entry. Use the following SQL statement to query the object:

```
SELECT * FROM HLQ.OSM_OBJ_DIR
WHERE   ODNAME = 'object_name';
```

Verify that the pending action date (ODPENDDT) is the current date or earlier. An object with a pending action date that is assigned a date in the future will not be selected for processing. For example, if today's date is 01/05/1999, and the pending action date for the object is 01/05/1999 or earlier, the object should be chosen for OSMC processing. If the pending action date for the object is 03/05/1999, the object is not chosen for OSMC processing until today's date is the same or earlier than the pending action date for the object.

Check the expiration date (ODEXPDT). If the object's expiration date has been reached, OSMC has no reason to move it to another level in the hierarchy. For more information concerning expiration dates, see "Processing Object Expiration" on page 202.

Collection Entry Not Found in the Collection Table

Another cause for nonselection of an object during OSMC processing may be that the collection entry for the collection ID associated with the storage group being processed cannot be found in the collection table of the object directory.

To verify that the collection table entry exists for the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_COLLECTION_TBL C, HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODCLID = C.ODCLID;
```

One row should be returned from the collection table identifying the collection that is expected to be associated with the object. If there is no collection table entry, the object is not selected for processing.

Inconsistencies between the Catalog and the Collection Table

Another possibility is a mismatch between the collection ID in the catalog and the collection ID in the collection table. OAM checks for this condition when the OAM address space is started. Check your system log during OAM address space initialization for message CBR9030I or other messages that may indicate a problem with the catalog.

The following TSO command lists the collection entry in the catalog:

```
LISTCAT ENTRIES('collection_name') ALL
```

Verify that the DIRECTORYTOKEN matches the storage group name for qualifier GROUP*xx* on the ISMF storage group definition panel.

Indicate "ACTIVE" as the SCDS name when listing Object storage group information. The storage group name in the catalog entry must also match the storage group name (ODCLSGNM) in the collection table entry for the collection associated with the object.

Check Management Class Definitions

To determine the storage class and management class names associated with the object, use the following SQL statement:

```
SELECT * FROM OAMADMIN.CBR_STO_CLASS_TBL S,
         OAMADMIN.CBR_MGT_CLASS_TBL M,
         HLQ.OSM_OBJ_DIR D
WHERE   ODNAME = 'object_name'
AND     D.ODSCNUM = S.ODSCNUM
AND     D.ODMCNUM = M.ODMCNUM;
```

If no rows are produced from the select statement, the object is not processed.

List the ISMF definition of the management class name returned from the select statement. This is the management class currently assigned to the object. Specify `ACTIVE` as the SCDS name.

Calculate the date of the next class transition as follows:
- If TIME SINCE CREATION has been used, add the values for time since creation to the creation date of the object (ODCREATS in the object's directory table entry).
- If TIME SINCE LAST USE has been specified, add the values for time since last use to the last referenced date of the object (ODLREFDT in the object's directory table entry).
- If PERIODIC has been specified, refer to the period definitions in the *z/OS DFSMSdfp Storage Administration Reference*.

In all cases, if the date calculated is in the future, no class transition occurs.

Check Management Class and Storage Class ACS Routines

If the ISMF management class definition indicates it is time for a class transition, the storage class ACS routine is invoked followed by the management class ACS routine. The ACS input variable &ACSENVIR will be set to CTRANS. In order for an object to be moved to a different level of the hierarchy, a new storage class must be assigned when &ACSENVIR is CTRANS.

Use the ISMF ACS test panels to determine what storage class is being assigned. Possible sources of error are as follows:
- CTRANS logic is not executed.
- The current storage class is assigned, or a new storage class is assigned, but the new storage class definition places the object at the same level of the hierarchy.

Check Storage Class Definitions

Display the ISMF storage class definition of the storage class assigned by the ACS routines. If INITIAL ACCESS RESPONSE SECONDS is zero, then the object is stored on DASD. If a nonzero value is specified, and SUSTAINED DATA RATE is greater than or equal to 3, then the object is stored on tape. Otherwise, the object is stored on optical media. If the media assigned by the storage class is unexpected, correct the storage class definition. If the wrong storage class is being assigned, correct the ACS routine. In either case, validate and activate the new configuration.

Documentation for Your IBM Representative

If you have not identified the reason for objects not moving to different levels of the hierarchy during OSMC, you may wish to contact the IBM support center. Please have the following documentation available:

1. Output from the select statements above.
2. The contents of the following tables:
   - OAMADMIN.CBR_MGT_CLASS_TBL
   - OAMADMIN.CBR_STO_CLASS_TBL, and
   - OAMADMIN.CBR_COLLECTION_TBL.
3. The contents of an object directory table entry for one of the objects expected to make a class transition.
4. The collection associated with the object directory table entry printed in step 3. The following command provides this information to you:

   ```
   LISTCAT ENTRIES('collection_name') ALL
   ```

5. A screen print of the storage group, storage class and management class definitions using “ACTIVE” as the SCDS name.
6. Management class and storage class ACS routines.
7. The syslog from the time OSMC message CBR9200I is issued to indicate the start of processing for the Object storage group until message CBR9201I is issued indicating processing has completed.

Invoking the OSREQ Macro Through the OSREQ TSO Command Processor

The OSREQ command is a TSO command processor that closely resembles the OSREQ macro, a programming interface, provided by OAM. The OSREQ macro is used for the storage, retrieval, query, deletion of objects, and comparing primary data to backup data. The OSREQ TSO command is used to exercise the OSREQ macro interface and OAM, without having to explicitly use the OSREQ ACCESS and UNACCESS macros to connect and disconnect the macro to OAM.

Each time the OSREQ command is issued, the OSREQ TSO command processor performs the OSREQ ACCESS macro between itself and OAM. If the OSREQ ACCESS macro is successful, the OSREQ command processor continues and performs the requested function. After the function is performed, the OSREQ command processor disconnects itself from OAM through a OSREQ UNACCESS macro command.

The following functions can be performed against objects: store, retrieve, query, delete, compare, and change the management class, storage class, and retention period.

Note: This command processor verifies object support after product installation. You may also use it as a tool to assist in recreating a customer problem in a controlled environment. You cannot use this tool to store actual data; the STORE function creates “dummy” data.

OSREQ TSO Command Syntax

The syntax for the OSREQ TSO command follows:

The OSREQ TSO Command Syntax

```
                                                 ┌─COMPARE───┐  ┌─NODATA─┐
►►──OSREQ──function collection-name object-name──┼───────────┼──┼────────┼────────►
                                                 └─NOCOMPARE─┘  └─DATA───┘

►──┬──────────────────┬──┬──────────────────┬──────────────────────────────────────►
   └─LENGTH(length)───┘  └─OFFSET(offset)───┘

►──┬──────────────────────────────────┬──┬────────────────────────────────────────┬──►
   └─MANAGEMENTCLASS(management-class)┘  └─RETENTIONPERIOD(retention-period)──────┘

►──┬────────────────────────────┬──┬─────────────────────┬─────────────────────────►◄
   └─STORAGECLASS(storage-class)┘  │        ┌─PRIMARY─┐  │
                                   └─VIEW(──┼─────────┼─)┘
                                            ├─BACKUP──┤
                                            └─BACKUP2─┘
```

The OSREQ TSO command processor requires three positional parameters:

1. The first positional parameter is the *function* to be performed.
2. The second positional parameter is the *collection-name*.
3. The third positional parameter is the *object-name*.

All three positional parameters are required and must be supplied in the order specified. If any of the positional parameters are missing, the system prompts the user for the missing parameter.

The first positional parameter following the OSREQ command name is the OSREQ function to be performed. Valid OSREQ functions are described in Table 37, and each one is discussed in detail further in this section.

Table 37. Functions of the OSREQ macro

FUNCTION	DESCRIPTION
CHANGE	The OSREQ CHANGE macro is invoked to change the management class, or retention period, or storage class, or all three, associated with the specified object.
COMPARE	The OSREQ RETRIEVE macro is invoked to retrieve both the primary object and the backup copy. The primary copy data is then compared with the backup data to insure the data is the same.
DELETE	The OSREQ DELETE macro is invoked to delete the specified object.
QUERY	The OSREQ QUERY macro is invoked for the specified collection name and object name.

Table 37. Functions of the OSREQ macro (continued)

FUNCTION	DESCRIPTION
RETRIEVE	The OSREQ RETRIEVE macro is invoked to retrieve the specified object.
STORE	The OSREQ STORE macro is invoked to store an object with the specified collection name and object name.

CHANGE

An OSREQ CHANGE command results in an OSREQ CHANGE macro invocation to change the management class, retention period, and or storage class associated with the specified object. The following optional keywords are valid for an OSREQ CHANGE command:

- MANAGEMENTCLASS
- RETENTIONPERIOD
- STORAGECLASS

Although there are no *required* keywords for an OSREQ CHANGE command, if none of the optional keywords are specified, the OSREQ CHANGE macro that is issued by the OSREQ TSO command processor has no affect on any of these keyword attributes currently associated with the object.

New attributes are assigned to the object dependent upon the attributes indicated on the keywords associated with the OSREQ CHANGE command.

COMPARE

An OSREQ COMPARE command results in the issuance of the OSREQ RETRIEVE macro to retrieve the primary and backup copies of the specified object. The data for the primary object then is compared with the data from the backup copy.

There are no required keywords on an OSREQ COMPARE command.

The following optional keywords are valid on an OSREQ COMPARE command:

- DATA or NODATA
- LENGTH
- OFFSET

The DATA keyword causes the DUMP notation display of the actual data comprising the object. Each line of object data consists of a message id, offset within the object, and 16 bytes of object data in both hexadecimal notation and EBCDIC format.

The LENGTH keyword specifies the length of the object, or portion thereof, for comparison. If the LENGTH keyword is omitted on an OSREQ COMPARE command, the OSREQ command processor issues an OSREQ QUERY macro for the specified object in order to obtain the length of the object. The portion of the object compared is also affected by the OFFSET keyword. If both the LENGTH and the OFFSET keywords are omitted, the entire object is compared using the length returned from the OSREQ QUERY macro. If the LENGTH keyword is supplied and the OFFSET keyword is omitted, the number of bytes specified with the LENGTH keyword starting at offset zero is compared.

The OFFSET keyword specifies the offset of the first byte of the object to be compared by the OSREQ COMPARE command. The first byte of an object has an offset of zero (0). The second byte of an object has an offset of one (1), and so on. If the OFFSET keyword is omitted from an OSREQ COMPARE command, the portion of the object compared is the portion staring with the first byte of the object (OFFSET=0). The number of bytes compared is specified with the LENGTH keyword.

Note: The compare function of the OSREQ command is different from the compare keyword on the retrieve function.

DELETE

An OSREQ DELETE command results in an OSREQ DELETE macro invocation to delete a specified object. There are no required or optional keywords for this command.

QUERY

An OSREQ QUERY command results in an OSREQ QUERY macro invocation for the specified collection name and object name. There are no required or optional keywords for this command.

You may specify the object name for the OSREQ QUERY command as a generic object name by specifying an asterisk (*) for the low-level qualifier of the object name. In the case where the low-level qualifier of the object name is an asterisk, the OSREQ QUERY macro that is issued is a “generic query” that may result in the directory information of multiple objects being displayed. In the case of a generic query, the query buffer obtained by the OSREQ TSO command processor is sufficient to hold the directory information for 10 000 objects. If there are more than 10 000 objects, only the directory information for the first 10 000 objects is listed, and the OSREQ TSO command processor ends with a return code four.

The following OAM directory information is listed for each object through the OSREQ TSO command processor:
- Collection name
- Object name
- Creation date
- Creation timestamp
- Last reference date
- Expiration date
- Storage class
- Management class
- Object length

RETRIEVE

An OSREQ RETRIEVE command results in an OSREQ RETRIEVE macro invocation to retrieve specified objects. There are no required keywords on this command.

The following optional keywords are valid on an OSREQ RETRIEVE command:
- DATA or NODATA
- COMPARE or NOCOMPARE
- LENGTH
- OFFSET
- VIEW(PRIMARY | BACKUP | BACKUP2)

The DATA keyword displays the actual data comprising the object in a DUMP notation. Each line of object data consists of a message ID, offset within object, 16-bytes of object data is both hexadecimal notation and EBCDIC format. The NODATA keyword suppresses the display of object data on the OSREQ RETRIEVE command. NODATA is the default if no keyword is specified on this command.

The COMPARE keyword checks the portion of the object retrieved for a predefined pattern of data. The predefined pattern is a combination of the collection name, object name and the 4-byte binary counter. If the portion of the object retrieved contains the expected data, a comparison successful message (CBR0420I) is issued. If the portion of the object retrieved does not contain the expected data, a comparison unsuccessful message (CBR0421I) is issued and the OSREQ TSO command processor ends with a nonzero return code. The NOCOMPARE keyword suppresses the checking of the predefined pattern. COMPARE is the default if no keyword is specified with this command.

The LENGTH keyword specifies the length of the object, or portion of the object to be retrieved. If the LENGTH keyword is omitted on the OSREQ RETRIEVE command, the OSREQ TSO Command Processor issues and OSREQ QUERY macro for the specified object to obtain the length of the object. The portion of the object retrieved is also affected by the OFFSET keyword. If both the LENGTH and the OFFSET keywords are omitted from the command, the entire object is retrieved using the length provided by the OSREQ QUERY macro. If the LENGTH keyword is supplied and the OFFSET keyword is omitted, the number of bytes specified with the LENGTH keyword, starting at offset zero, is retrieved.

The OFFSET keyword specified the offset of the first byte of the object retrieved by the OSREQ RETRIEVE command. The first byte of an object has an offset of zero. The second byte of an object has an offset of one and so on. If the OFFSET keyword is omitted from the command, the portion on the object retrieved is the portion starting with the first byte of the object (OFFSET=0). The number of bytes retrieved is specified with the LENGTH keyword.

The VIEW keyword is only valid on an OSREQ RETRIEVE request and is ignored on all other requests. If you specify VIEW(PRIMARY) on the RETRIEVE request, the primary copy of the object is retrieved. If the primary copy of the object is on an unreadable tape or optical volume, and the automatic access to backup facility is activated, OAM then retrieves the backup copy of the object. If you specify VIEW(BACKUP) on the RETRIEVE request, then OAM retrieves the first backup copy of the object. If you specify VIEW(BACKUP2) on the RETRIEVE request, then OAM retrieves the second backup copy of the object. If you omit the VIEW keyword, then OAM uses VIEW(PRIMARY) as the default in the request and the retrieves the primary copy of the object.

STORE

An OSREQ STORE command results in an OSREQ STORE macro invocation to store an object with a specified collection name and object name.

The LENGTH keyword is a required keyword on an OSREQ STORE command. The value specified with the length keyword is the length, in bytes, of the object to store. If the LENGTH keyword or value is omitted, an error message is issued.

The OSREQ command processor creates an object of the specified length and fills the object with a predefined pattern. The predefined pattern is a combination of the collection name, object name and the 4-byte binary counter. The binary counter in the data pattern is incremented by one for each replication of the data pattern within the object.

The following keywords are valid on an OSREQ STORE command:
- LENGTH (required)
- MANAGEMENTCLASS (optional)

- RETENTIONPERIOD (optional)
- STORAGECLASS (optional)

New attributes are assigned to the object dependent upon the attributes indicated on the keywords associated with the OSREQ STORE command. If these attributes are not specified, the defaults for the collection or the ACS overrides are assigned.

The second positional parameter, following the OSREQ command name is the *collection name*. The collection name must be a fully qualified OAM collection name for each OSREQ function.

The third positional parameter to follow the OSREQ command name is the *object-name*. The object name should be a fully-qualified object name for each OSREQ function with the exception of the QUERY function. For the OSREQ QUERY function, the object name can be either a fully-qualified or a partially-qualified name. A partially-qualified name is an object name that contains one asterisk (*) for the right most portion of the object name. A partially-qualified object name implies a generic-query request for all objects whose object name matches the characters to the left of the asterisk.

OSREQ TSO Command Processor Return Codes

The OSREQ TSO command processor returns to the TSO terminal monitor program (TMP) with a return code in register 15. This return code can be tested using the &LASTCC variable in a TSO CLIST. In all cases, except one, the return code in register 15 following the OSREQ command is the return code that was returned by the OSREQ macro in register 15. A return code of 20 indicates that the OSREQ TSO command processor encountered an error unrelated to the OSREQ macro which it invokes.

The following return codes are returned from the OSREQ TSO command processor:

Return code **Description**

0
The requested OSREQ function successfully completed.

4
The requested OSREQ function was completed with a warning condition.

8
The requested OSREQ function was not completed due to an application programming error.

12
The requested OSREQ function was not completed due to an environmental error.

20
The OSREQ TSO command processor encountered an error during its processing. The following errors will cause a return code of 20:

Error	**Description**
1	A nonzero return code was received from the TSO parse service routine (IKJPARS) in register 15. Error message CBR0402I is issued.
2	A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ QUERY operation. Error message CBR0403I is issued.
3	The LENGTH keyword was not specified on an OSREQ STORE request. Error message CBR0406I is issued.

4 A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ STORE operation. Error message CBR0403I is issued.

5 A nonzero return code was received from the STORAGE OBTAIN macro when storage was requested for a data buffer for an OSREQ RETRIEVE operation. Error message CBR0403I is issued.

6 An invalid length was specified on an OSREQ RETRIEVE or COMPARE command. Error message CBR0407I is issued.

7 An invalid offset was specified on an OSREQ RETRIEVE or COMPARE command. Error message CBR0408I is issued.

8 An OSREQ RETRIEVE command was issued with the COMPARE keyword and the data comparison did not match. Or an OSREQ COMPARE command was issued and the data comparison did not match. Error message CBR0421I is issued.

3995 Optical Service Information Messages

The 3995 optical library microcode provides a 3995 Service Information Message (SIM) to the host when a component within the 3995 optical library needs service.

If the appropriate bit is on in the SIM, OAM issues an action message (CBR3309E) to the MVS console. Additionally, if the appropriate bit is on in the SIM, OAM logs the SIM message as a type x‘A3’ Asynchronous Notification Record (ANR) in SYS1.LOGREC.

The following restrictions apply:

- The 3995 optical library dataservers must be defined in the IODF as 3995 devices (not CTC or 3088 devices) in order for the Environmental Recording Editing and Printing program (EREP) to produce formatted reports for any ANRs generated for those libraries.
- AMRF must be active in order to recall CBR3309E messages that have rolled off the screen. To display the message identification numbers and text of all immediate action and eventual action messages, as well as OAM issued messages awaiting replies, use the following command:

```
DISPLAY R,L,KEY=OAM
```

- OAM must be active when SIM attentions from the 3995 are sent to the host or they are lost.
- The optical libraries indicated in the SIM attention messages must be known to OAM, and they must be part of the active SMS configuration or they are lost.

Chapter 5. Operating OAM and OTIS Address Spaces and OSMC Functions

This chapter helps you use operator commands and describes the tasks to operate the OAM and OTIS (necessary for the support of OSREQ functions) address spaces and OSMC functions.

Message Format Conventions

In this chapter, the following conventions are used to show message format:

```
CBRnnnnX Message_text
```

where:

CBR Standard OAM message prefix

nnnn Four-digit message number

X Type code
- **A** Action required
- **D** Decision needed
- **E** Eventual action required
- **I** Information only

Message_text
Text of the message

For a description of messages, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

The following is a sample of an OAM message:

```
CBR2601A Specify shelf location for volume volser.
```

Note: In message text, italicized words indicate a value supplied by the system.

Overview of Operator Commands

Many commands use the MVS operator MODIFY command. Throughout this topic, the command syntax is:

```
F OAM,...
```

The command F is the abbreviation of the MVS MODIFY command.

The OAM address space can be defined in one of the following ways:

- As the name of the cataloged procedure in SYS1.PROCLIB that you use to start the OAM address space:

```
F OAM,START OAM,...
```

- As the task ID assigned in the address space START command:

```
F OAM,START procname.OAM,...
```

Note: If your system programmer chooses to use a name other than *OAM*, use that name in the place of *OAM*.

Two SMS operator commands are related to the OAM address space:
- **DISPLAY SMS**—for determining the status of OAM, OSMC, storage groups, optical volumes, optical disk drives, and optical libraries. See "Displaying Status" on page 268 for specific information on the DISPLAY command.
- **VARY SMS**—for varying optical disk drives or optical libraries online or offline. See "Varying Optical Drives and Libraries" on page 244 for specific information on the VARY command.

The following operator commands for the OAM address space are based on the MVS MODIFY command:
- **AUDIT**—for auditing library resident volumes. See "Auditing a Volume" on page 317 for specific information on the AUDIT command.
- **DISPLAY**—for displaying the current SETOAM, SETOPT, and SETOSMC settings. See "Displaying SETOPT, SETOAM, and SETOSMC Statements" on page 302 for specific information on this use of the DISPLAY command.
- **DUMP**—for capturing data used for diagnostic purposes. See "Scheduling an SVC Dump for the OAM Address Space" on page 308 for specific information on the DUMP command.
- **LABEL**—for labeling the two volumes of an optical disk on a stand-alone or operator-accessible optical drive. See "Labeling an Optical Disk on a 9247 Stand-Alone Drive" on page 256 and "Labeling an Optical Disk on a 3995 Operator-Accessible Drive" on page 259 for specific information on the LABEL command.
- **QUERY**—for displaying summary, detail, or both types of information regarding active and pending tape and optical requests. See "Querying Summary and Detail Information for Pending and Active Requests" on page 304 for specific information on the QUERY command.
- **RELABEL**—for changing the volume serial number for an existing optical disk volume. See "Relabeling a 3995 Optical Disk Volume" on page 264 for specific information on the RELABEL command.
- **REMAP**—for initiating a REMAP on a 3995 optical library. See "Remapping an Optical Library" on page 318 for specific information on the REMAP command.
- **RESTART**—for restarting the OAM address space without issuing a STOP and START command. See "Restarting the OAM Address Space" on page 309 for specific information on the RESTART command.
- **START**—for starting an OSMC storage management cycle, an OSMC library space management cycle, an OSMC DASD space management cycle, an OSMC Volume Recovery utility, an OSMC Single Object Recovery utility, an OSMC Move Volume utility, or the automatic access to backup function. Consult the following references for specific information on the START command:
  – "Starting OAM" on page 221

- “Starting OSMC Functions” on page 223
- “Starting Automatic Access to Backup Copies of Objects” on page 241

- **STOP**—for stopping OAM, all OSMC processing, OSMC processing for an individual storage group, an OSMC Move Volume utility, or processing for the access backup recovery function. See “Stopping OAM Functions” on page 318 for specific information on this STOP command.
- **UPDATE**—for updating fields in the DB2 Volume Table and the Tape Volume Table with the F OAM command. This command is also used to dynamically update the SETOAM, SETOPT, and OAMXCF settings. Consult the following references for specific information on the UPDATE command:
  - “Using the UPDATE Command to Set SETOAM and SETOPT Values” on page 311
  - “Using the UPDATE Command to Set OAMXCF Values” on page 314
  - “Updating Fields in the DB2 Volume Table and the Tape Volume Table” on page 315

The following operator commands for the OAM address space are based on the MVS LIBRARY command:

- **EJECT**—for removing an optical disk from an optical library. See “Ejecting an Optical Disk” on page 253 for specific information on the EJECT command.
- **RESET**—for re-enabling the CBRUXSAE user authorization installation exit after it has been disabled or bypassed.

Two MVS operator commands are related to the OTIS address space:

- **START**—for starting or restarting OTIS. See “Starting OTIS” on page 221 for specific information on this use of the START command.
- **STOP**—for stopping OTIS. You must use the MVS MODIFY command when stopping OTIS. See “Stopping OTIS” on page 321 for specific information on this use of the STOP command.

Overview of Operator Tasks

You can perform the following tasks:

- Start:
  - OTIS
  - OAM
  - OSMC
    - Storage management cycle for all Object storage groups
  - Storage management cycle for a specific Object storage group
  - Library management cycle
  - DASD space management cycle
  - Automatic Access to Backup
- Recover an optical disk or tape volume:
  - Object storage group
  - Object Backup storage group
- Recover a single object from optical or tape.
- Access backup copy of an object automatically.
- Move objects off of a source volume (primarily for media migration).
- Vary:
  - Optical drive online and offline
  - Optical library online and offline
- Enter:

- Unlabeled optical disk into an optical library
  - Labeled optical disk into an optical library
- Eject an optical disk from an optical library.
- Mount an optical disk on a stand-alone or operator-accessible optical drive.
- Demount and remove an optical disk cartridge from a stand-alone or operator-accessible drive.
- Label an optical disk on a stand-alone or operator-accessible optical drive.
- Relabel an optical disk volume.
- Display status of:
  - OAM
  - OSMC and OSMC functions
    - Storage management cycle for all Object storage groups
    - Storage management cycle for a specific Object storage group
    - DASD space management
    - Library space management
    - Volume Recovery utility
    - Move Volume utility
  - Optical drive
  - Optical library
  - Storage group
  - Optical volume
  - OAM XCF
  - SETOAM, SETOPT, and SETOSMC parameters
  - Lost volumes
- Display Outstanding OAM messages
- Query:
  - To display summary and detail information regarding active and pending optical and tape requests.
- Dump:
  - To provide diagnostic data about the OAM address space.
- Restart OAM:
  - To restart OAM without performing a STOP and START command.
- Update SETOAM, SETOPT, and OAMXCF parameter values.
- Update fields in the DB2 VOLUME or TAPEVOL Table
- Stop:
  - OAM
  - OSMC and OSMC functions
    - Storage management cycle for a single storage group
    - Move Volume utility for an optical volume
  - Access Backup
  - OTIS
- Remap a 3995 library
- Audit
  - Single volume
  - Volume list
  - all volumes in a library

Starting OTIS

The installation of OAM causes the creation of an address space called OAM thread isolation support (OTIS). This required address space starts automatically during system IPL.

Note: The OTIS address space must be active when any OAM applications are processing objects.

To start the OTIS address space or restart the address space after it has ended, enter the MVS START command. The following MVS command syntax starts the OSMC functions:

```
►►──START──OTIS──────────────────────────────────────────────────────►◄
```

For further information on the MVS START command, refer to the *z/OS MVS System Messages, Vol 4* .

While OTIS is initializing, the system issues the following message:

```
CBR8500I OTIS subsystem is initializing.
```

If your DB2 subsystem is active, the system issues the following message:

```
CBR8571I OTIS subsystem successfully connected to DB2 subsystem.
```

If your DB2 subsystem is *not* active, the system issues the following message:

```
CBR8572I OTIS subsystem unable to connect to DB2 subsystem because DB2
         subsystem is not active.
```

After either of the above messages are issued, the system issues the following message:

```
CBR8501I OTIS subsystem initialization complete.
```

You should not see any messages other than those listed. If any other messages are issued, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Starting OAM

Before starting OAM for object support, initialize DB2. For information on starting DB2, refer to *DB2 for OS/390 Administration Guide*.

The system programmer can update the SMS entry in the IGDSMS*xx* member of PARMLIB to automatically start OAM during MVS IPL.

To start the OAM address space manually, or to restart the OAM address space after it has terminated, enter the MVS START command. The following syntax of the MVS START command starts OAM:

```
►►──START──OAM──,──┬───────────────┬──┬────────────────────────┬──────────────►◄
                   └─OAM──=──xx──,─┘  └─parameter──=──value─┘
```

Where *xx* is the CBROAM*xx* member of PARMLIB that you wish to have OAM use during initialization.

Note: Any parameter of the OAM procedure statement can be included in this command to modify the parameter upon activation of the OAM address space. An example of modifying these parameters follows:

```
START OAM,OAM=XX,OSMC=YES,RESTART=NO,MAXS=10,EJECT=LRM
```

For details on these parameters, see the discussion concerning modifying and running the CBRAPROC SAMPLIB member on page 114. For further information on the MVS START command, refer to *z/OS MVS System Commands*.

When initializing OAM with OSMC, the system issues the following messages:

```
CBR0001I OAM initialization starting.

CBR0016I Successful processing of the OAMXCF commands in CBROAMxx member of
         PARMLIB.  Initialization continues.

CBR0070I OAM XCF member xcf-member-name is the first member defined to
         OAM XCF group xcf-group-name, group successfully defined to
         XCF and member created.

CBR0016I Successful processing of the SETOAM commands in CBROAMxx member of
         PARMLIB. Initialization continues.

CBR0016I Successful processing of the SETOPT commands in CBROAMxx member of
         PARMLIB. Initialization continues.

CBR0016I Successful processing of the SETOSMC commands in CBROAMxx member of
         PARMLIB. Initialization continues.

CBR9000I OSMC initialization starting.

CBR9001I OSMC initialization completed.

CBR0002I OAM initialization completed.
```

Notes:

1. CBR0016I displays only if OAM=*xx* is indicated.
2. If you use a CBROAM*xx* PARMLIB member to initialize OAM, there are CBR03*xx*I messages that may be normal for your environment. Refer to *z/OS MVS System Messages, Vol 4* for explanations and system actions for these messages.
3. OAM must be restarted to recognize any changes that were made to the CBROAM*xx* member of PARMLIB. Additionally, to dynamically change the SETOAM or SETOPT parameters, use the UPDATE command. See "Using the UPDATE Command to Set SETOAM and SETOPT Values" on page 311 for more information.

When initializing OAM without OSMC, the system issues the same messages as when initializing with OSMC, with the exception of messages CBR9000I and CBR9001I.

For further information on these messages, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Starting OSMC Functions

Use the OAM START command to start the following OSMC functions:

- Storage management cycle for all Object storage groups or an individual Object storage group
- Library space management cycle for an optical library
- DASD space management cycle for an Object storage group
- Volume Recovery utility
- Single Object Recovery utility
- Move Volume utility

The following MVS command syntax starts the OSMC functions:

```
►►──┬─MODIFY─┬──────────────────────────────────────────────────────────►
    └─F──────┘

►─OAM,──┬─START─┬─,──┬─OSMC──────────────────────────────────────┬──────►◄
        └─S─────┘    ├─STORGRP,storage-group-name────────────────┤
                     ├─LIBMGT,library-name───────────────────────┤
                     ├─DASDSM,storage-group-name─────────────────┤
                     │                       ┌─BACKUP1─┐         │
                     ├─RECOVERY,volser,──────┤         ├─────────┤
                     │                       └─BACKUP2─┘         │
                     │                                 ┌─BACKUP1─┐
                     ├─OBJRECV,collection-name,object-name,─┤   ├┤
                     │                                 └─BACKUP2─┘
                     └─MOVEVOL,volser────────────────────────────┘
```

OSMC
: Starts an OSMC storage management cycle. If no additional parameters are provided, the management cycle runs for *every* Object storage group in the active configuration. In support of an OAMplex, OAM will initiate OSMC processing for object storage groups that have the current system name specified (or no system name specified) in the object storage group definition as the system where OSMC processing is to take place.

STORGRP,*storage-group-name*
: Starts an OSMC storage management cycle for the specified Object storage group that is named *storage-group-name*. In an OAMplex environment, if a specific storage group is requested to be processed, OAM initiates OSMC processing of the specified storage group on the system where the command was entered.

LIBMGT,*library-name*
: Starts an OSMC library management cycle for an optical library that is named *library-name*. This command requires that the library that is requested for library space management processing is currently managed and controlled by the OAM on which the command was entered.

DASDSM,*storage-group-name*
Starts an OSMC DASD space management cycle for the specified Object storage group that is named *storage-group-name*.

RECOVERY,*volser***,BACKUP1 | BACKUP2**
Starts the OSMC Volume Recovery utility for either a tape volume with volume serial number *volser* or an optical volume with volume serial number *volser* and its opposite side. When all objects residing on a volume belonging to an Object storage group are to be recovered, and BACKUP1 is specified on the START command for volume recovery, the recovery is made from the first backup copies of the objects. When all objects residing on a volume belonging to an Object storage group are to be recovered, and BACKUP2 is specified on the START command for volume recovery, the recovery is made from the second backup copies of the objects. If the volume or cartridge to be recovered belongs to an Object storage group and neither BACKUP1 nor BACKUP2 is specified, then BACKUP1 is the default. Backup volumes are always recovered using the primary copies of the objects.

OBJRECV,*collection-name***,***object-name***,BACKUP1 | BACKUP2**
Starts the OSMC Single Object Recovery utility for the object that is named *object-name*, in the collection that is named *collection-name*. When a primary copy of an object is to be recovered, and BACKUP1 is specified on the START command for object recovery, the recovery is made from the first backup copy of the object. If BACKUP2 is specified, the recovery is made from the second backup copy of the object. BACKUP1 is the default if neither BACKUP1 nor BACKUP2 is specified.

MOVEVOL,*volser*
Starts the OSMC Move Volume utility for the source volume with volume serial number *volser*. This command requires that the volume that is requested for the Move Volume utility is either:

- Currently managed and controlled by the OAM on which the command was entered
- Or, the volume is shelf-resident and not currently mounted

Starting the Storage Management Cycle

The OAM START command can be used to start the storage management cycle for all Object storage groups in the active configuration or for an individual Object storage group.

Starting the Storage Management Cycle for All Storage Groups

To start the OSMC storage management cycle for all Object storage groups:

1. Enter the following command:

```
F OAM,START,OSMC
```

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

3. The system issues the following messages pertaining to the storage management cycle as seen on the console.

```
CBR9018I OSMC starting Storage Management Cycle.

CBR9200I Object Processing starting for storage group xxxxx01.

CBR9370I OSMC Detail for taskname:

          READ   READ   READ  WRITE  WRITE
          DASD    OPT   TAPE   DASD    OPT
WORK Q: aaaaaa bbbbbb cccccc dddddd eeeeee
WAIT Q:        jjjjjj kkkkkk        llllll
DONE:   qqqqqq rrrrrr ssssss tttttt uuuuuu

         WRITE   WRITE   WRITE    DIR
          TAPE BACKUP1 BACKUP2  UPDTS
WORK Q: ffffff  gggggg  hhhhhh iiiiii
WAIT Q: mmmmmm  nnnnnn  oooooo pppppp
DONE:   vvvvvv  wwwwww  xxxxxx yyyyyy
End of Display Detail

CBR9201I Object Processing completed for storage group xxxxx01.

CBR9500I Shelf Manager starting for storage group xxxxx01.

CBR9501I Shelf Manager completed for storage group xxxxx01.  0 optical disks
         selected.

CBR9048I Storage Group xxxxx01 has successfully completed processing.

CBR9200I Object Processing starting for storage group xxxxx12.

CBR9370I OSMC Detail for taskname:

          READ   READ   READ  WRITE  WRITE
          DASD    OPT   TAPE   DASD    OPT
WORK Q: aaaaaa bbbbbb cccccc dddddd eeeeee
WAIT Q:        jjjjjj kkkkkk        llllll
DONE:   qqqqqq rrrrrr ssssss tttttt uuuuuu

         WRITE   WRITE   WRITE    DIR
          TAPE BACKUP1 BACKUP2  UPDTS
WORK Q: ffffff  gggggg  hhhhhh iiiiii
WAIT Q: mmmmmm  nnnnnn  oooooo pppppp
DONE:   vvvvvv  wwwwww  xxxxxx yyyyyy
End of Display Detail

CBR9201I Object Processing completed for storage group xxxxx12.

CBR9500I Shelf Manager starting for storage group xxxxx12.

CBR9501I Shelf Manager completed for storage group xxxxx12.  0 optical disks
         selected.

CBR9048I Storage Group xxxxx12 has successfully completed processing.

CBR9009I OSMC completed its Storage Management Cycle.  12 tasks started.
         12 tasks completed.
```

If errors occur during the storage management process, additional messages may be issued, such as message CBR9049I to indicate unsuccessful completion.

Starting the Storage Management Cycle for an Individual Storage Group

To start the OSMC storage management cycle for an individual Object storage group:

1. Enter the following command:

```
F OAM,START,STORGRP,storage-group-name
```

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

3. The system issues the following set of messages for the storage group:

```
CBR9200I Object Processing starting for storage group storage-group-name.

CBR9370I OSMC Detail for taskname:

          READ   READ   READ  WRITE  WRITE
          DASD    OPT   TAPE   DASD    OPT
WORK Q: aaaaaa bbbbbb cccccc dddddd eeeeee
WAIT Q:        jjjjjj kkkkkk        llllll
DONE:   qqqqqq rrrrrr ssssss tttttt uuuuuu

         WRITE   WRITE   WRITE    DIR
          TAPE BACKUP1 BACKUP2  UPDTS
WORK Q: ffffff  gggggg  hhhhhh iiiiii
WAIT Q: mmmmmm  nnnnnn  oooooo pppppp
DONE:   vvvvvv  wwwwww  xxxxxx yyyyyy
End of Display Detail

CBR9201I Object Processing completed for storage group storage-group-name.

CBR9500I Shelf Manager starting for storage group storage-group-name.

CBR9501I Shelf Manager completed for storage group storage-group-name.
         n optical disks selected.

CBR9048I Storage Group storage-group-name has successfully completed processing.
```

If errors occur during the storage management process for a storage group, additional messages may be issued, such as message CBR9049I to indicate unsuccessful completion.

Starting the Library Space Management Cycle

Starting the library space management cycle ejects the least-recently-written or the least-recently-mounted optical disk from the optical library. This command requires that the library requested for library space management processing is currently managed and controlled by the OAM on which the command was entered. If the member name associated with the optical library specified on this command is not the member name for this instance of OAM, the command fails and message CBR1068I is issued.

Note: If the Least-Recently-Mounted (LRM) parameter was specified on the EJECT keyword of the OAM procedure statement, the library space management cycle ejects the least-recently-mounted optical disk from the optical library.

The OSMC library space management cycle can be started:

- Manually, by entering the START command. Use the START command if the library is full or if scratch volumes are needed in the library.
- Automatically, when OAM cannot locate a library-resident optical volume on which to write an object and the following conditions are met:

– For the storage group:
  - There are no library optical volumes or
  - Those residing in the library are not usable at this time.
– There are no scratch volumes in the optical library.
– The library is full.

To start the OSMC library management cycle:

1. Enter the following command:

```
F OAM,START,LIBMGT,library-name
```

2. The system issues the following messages:

```
CBR1000I OAM START command execution scheduled.
CBR9400I Library Space Manager starting for library library-name.
```

3. If a shelf location has not been previously specified for the optical disk being ejected, the system issues the following message and waits for a reply:

```
CBR2600A Specify shelf location for volumes volser-1 and volser-2.
```

4. Provide 1 to 32 characters of shelf information.

   If the optical volume being ejected does not already have a valid pseudo library associated with it, or the library from which the volume was ejected does not have a default pseudo library in its definition, the system issues message CBR2602A or message CBR2603A, or both:

```
CBR2602A Eject pending for volser in r-library-name. Default pseudo
         library is p-library-name. Reply 'U' to use, or 'R' to respecify.
```

```
CBR2603A Specify pseudo library name for volume volser.
```

   If the volume being ejected has an invalid pseudo library associated with its volume record and the library from which the volume is being ejected has a default pseudo library in its SCDS definition, message CBR2602A is issued. This message asks if the default is to be used or if another pseudo library name is requested.

   If the library from which the volume is being ejected does not have a default pseudo library in its SCDS definition, or “R” was replied to message CBR2602A, then the system issues message CBR2603A. This message requests a pseudo library name to assign the volume to when it becomes shelf resident.

   If a pseudo library name specified in response to message CBR2603A is not valid in the current active control data set (ACDS), or the volume record has an invalid pseudo library name associated with it, the system issues message CBR2604I, which is followed by message CBR2603A:

```
CBR2604I Volume volser cannot be assigned to pseudo library p-library-name, it
         is not a valid pseudo library definition in the active SMS configuration.
CBR2603A Specify pseudo library name for volume volser.
```

The system puts the optical disk cartridge in the input/output station and issues the following messages:

```
CBR3122I Volumes volser-1 and volser-2 were ejected from library library-name
         shelf location is shelfloc.

CBR3001A Remove cartridge from I/O station on library library-name.  Place in
         shelf location shelfloc.
```

5. Remove the optical disk cartridge from the input/output station of the specified optical library and return it to the shelf location indicated.

```
CBR9401I Library Space Manager completed for library library-name.  n optical
         disks ejected.
```

If any errors occur during the library space management cycle, additional messages may be issued identifying the error, and message CBR9401I is not issued. Use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Starting the DASD Space Management Cycle for an Individual Storage Group

Use DASD space management to do the following:
- Select objects that require processing.
- Determine if they have expired by examining the objects’ explicit expiration date or examining the objects’ management class information.
- Physically delete data for expired objects on DASD and on re-writable optical disks.
- Remove the object directory entry (indicating volume serial number and sector location and other object information) for the expired object.

To start DASD space management:

1. Enter the following command:

```
F OAM,START,DASDSM,storage-group-name
```

2. The system issues the following messages:

```
CBR9300I DASD Space Management starting for storage group storage-group-name.

CBR9301I DASD Space Management completed for storage group storage-group-name.
```

3. The following message appears if OAM is having some major problems and cannot perform DASD space management on a storage group. This message would be preceded by other CBR*xxxx*I messages.

```
CBR9043I DASD Space Manager not started for storage-group-name.
```

If any errors occur during the DASD space management cycle, additional messages may be issued identifying the error, and message CBR9043I is not issued. For a description of this message, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Note: If OAM detects that an object storage table or an object directory is unable to extend its storage during normal processing of objects using the OSREQ application interface, DASD space management is automatically started for that storage group. If the table that runs out of extents is the object directory, all object stores to that storage group fail until more space is made available to the object directory for the storage group. DASD space management is started automatically in this instance so that subsequent stores to the storage group may have improved chances of storing successfully.

Starting Recovery Functions

OAM supports recovery functions at three levels:

- Recovering tape volumes or optical disks
- Recovering single objects
- Activating access to backup copy of objects

Starting the OAM Volume Recovery Utility

The OAM Volume Recovery utility recovers only objects that reside on an unusable optical or tape volume. It does not recover objects to DASD volumes. The utility only retrieves copies of the objects that are stored on DASD to recover them to optical or tape volumes when recovering an Object Backup volume. See “Recovering an Entire Optical Cartridge or Tape Volume” on page 192 for more information.

Typically, some of the objects are recovered to the Object Backup or Object storage group volume that is currently being written, and the rest of the objects are
| recovered to the next assigned Object Backup or Object storage group volume. If
| there are two backup copies of the object, you can indicate which copy of the object
| (first or second backup copy) will be used in the recovery by indicating BACKUP1 **|**
| BACKUP2 on the RECOVERY command.

Note: The system may issue a message requesting that a scratch volume be mounted during recovery.

| Once recovery is started and you reply **GO** to message CBR9810D, you can stop volume recovery by issuing one of the following commands:

F OAM,STOP,OSMC
: This command stops the OSMC process, and thereby, stops the volume recovery processing. It is recommended that this command be used to stop the volume recovery process.

F OAM,STOP,OAM
: This command stops *all* OAM processing, not just the volume recovery process. Caution should be used when issuing this command for that reason.

Starting Recovery Functions for an Optical or Tape Volume

To recover an optical or tape volume:

1. Enter the following RECOVERY command:

```
F OAM,START,RECOVERY,volser,BACKUP1 | BACKUP2
```

   where *volser* is the volume serial number of one of the volumes that is being recovered.

2. The system issues the following messages:

```
CBR1000I  OAM START command execution scheduled.

CBR9800I  OAM Volume Recovery starting for volumes volser-1 and volser-2.

CBR9827I  OAM Volume Recovery.  The following TAPE volumes are needed
          for recovery: volser-1 volser-2 volser-3 volser-4 volser-5
          volser-6 volser-7 volser-8 volser-9

CBR9824I  OAM Volume Recovery.  The following OPTICAL volumes are needed
          for recovery: volser-1 volser-2 volser-3 volser-4 volser-5
          volser-6 volser-7 volser-8 volser-9.
```

   Message CBR9800I indicates that volume recovery has started for volumes *volser-1* and *volser-2*.

   Note: In situations where a tape volume is being recovered, *volser-2* is labeled as **N/A**.

   Message CBR9824I gives you a list of optical volumes to retrieve that are identified by *volser-n*. This message allows you to get the optical volumes that are needed for recovery processing.

   Message CBR9827I gives you a list of tape volumes to retrieve that are identified by *volser-n*. This message allows you to get the tape volumes that are needed for recovery processing.

   To recover a primary volume, all of the backup volumes containing backup copies of the object on the primary volume are needed whether they are optical or tape. Also, to recover a backup volume, *every* Object storage group must be searched for objects which have a backup copy on the backup volume to be recovered. For each of these objects, the primary copy is used to recover the backup volume. The primary copy of these objects could be on DASD, optical, or tape. As a result, the Volume Recovery utility must identify the optical and tape volumes that are needed for the recovery.

   If more than one backup copy of the objects exists, you can select which backup copy will be used for the recovery (first or second backup copy) by specifying BACKUP1 | BACKUP2 on the F OAM,START,RECOVERY command.

3. For the volumes listed in messages CBR9827I and CBR9824I, the system issues the following message:

```
CBR9810D  Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
```

4. If all the volumes are not available, reply QUIT to terminate recovery, and start again when the volumes have been retrieved.

   If you reply QUIT to CBR9810D, the system issues the following message:

```
CBR9819I  OAM Volume Recovery is ending for volumes volser1 and volser2
```

5. If the volumes are available, reply GO to CBR9810D.
6. The system issues the following message for each optical volume listed in message CBR9824I:

```
CBR4400A  Mount volume volser on drive drive-name.  Shelf location is shelfloc.
```

7. The system issues the following message for each tape volume that is listed in the message CBRR9827I:

```
IEC501A  M drive-Addr,volser,label,,,,data_set_name.
```

   Message CBR9824I may identify volumes that are either library-resident or shelf-resident optical volumes. The system automatically mounts the library-resident optical volumes; therefore, a mount message is not issued for them. The mount message CBR4400A requests only shelf-resident optical volumes for recovery.
8. Mount the optical volume that is identified by *volser* in message CBR4400A and any tape volumes identified by volser in message IEC501A.

   When recovery is complete, the system issues the following message:

```
CBR9819I  OAM Volume Recovery is ending for volumes volser1 and volser2
```

The following is a sample of the F OAM,START,RECOVERY,*volser* command:

```
F OAM,S,RECOVERY,WG360A,BACKUP2

CBR1000I OAM S command execution scheduled.
CBR9800I OAM Volume Recovery starting for volumes WG360A and WG360B.
CBR9827I OAM Volume Recovery.
The following TAPE volumes are needed for recovery:
 TNN005
CBR9824I OAM Volume Recovery.
The following OPTICAL volumes are needed for recovery:
 WG688A
CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.

If you reply GO, the system issues the following messages:

CBR9852I Volume Recovery Utility processing objects in storage group GROUP22
         for volume WG360A.
IEC501A M 1D04,TNN005,SL,COMP,OAM,OAM,OAM.BACKUP2.DATA
CBR4401I Volume WG830B mounted on drive P9D3.
CBR9863I Volume Recovery status for volumes WG360A and WG360B.  Total: 5, Attempted: 5,
         Successful: 5, Unsuccessful: 0.
CBR9819I OAM Volume Recovery is ending for volumes WG360A and WG360B.

If you reply QUIT, the system issues the following messages:

CBR9862I Volume Recovery status for volumes WG360A and WG360B is not available.
CBR9819I OAM Volume Recovery is ending for volumes WG360A and WG360B.
```

If any errors occur during the volume recovery process, additional messages may be issued identifying the errors. For a description of the messages, use LookAt or see *z/OS MVS System Messages, Vol 4*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Message CBR9863I has been added to provide statistics for volume recovery status. The statistics include: Total, Attempted, Successful, and Unsuccessful. These statuses have the following meanings:

Total The total number of objects found on *volser1* and *volser2*

Attempted The total number of objects for which processing has begun in this utility for *volser1* and *volser2*

Successful The total number of objects successfully recovered for *volser1* and *volser2* and written to other volumes

Unsuccessful The total number of attempted objects for which processing has begun in this utillity, but which were not successfully recovered for *volser1* and *volser2*

Note: This number does not necessarily mean that processing failed for these objects, but only that processing had started but was not completed.

The text of the message follows:

```
CBR9863I  Volume Recovery status for volumes volser1 and volser2.
          Total: total, Attempted: attempted, Successful: successful
          Unsuccessful: unsuccessful
```

Statistics Provided by the Volume Recovery Utility

Normal Completion of Recovery of a Backup Tape Volume: Figure 26 on page 233 shows recovery of a backup tape volume from the primary objects that reside on both tape and optical volumes. Recovery completion is normal. Twelve first backup copies of objects were identified as residing on the backup volume. All 12 backup objects are successfully recovered from the primary objects.

```
F OAM,S,RECOVERY,CMW008
CBR1000I OAM S command execution scheduled.
CBR9800I OAM Volume Recovery starting for volumes CMW008 and N/A.
CBR9827I OAM Volume Recovery.
         The following TAPE volumes are needed for recovery:
         CMW006
CBR9824I OAM Volume Recovery.
         The following OPTICAL volumes are needed for recovery:
         WG488A WG488B *38
CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
         R 38,GO
IEE600I  REPLY TO 38 IS;GO
CBR9852I Volume Recovery Utility processing objects in storage group
         GROUP28 for volume CMW008.
IEC501A  M 1D18,CMW006,SL,COMP,OAM,OAM,OAM.PRIMARY.DATA
IEC501A  M 1D19,CMW004,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
CBR4401I Volume WG488A mounted on drive P15D2.
CBR4401I Volume WG488B mounted on drive P15D2.
CBR9863I Volume Recovery status for volumes CMW008 and N/A. Total: 12,
         Attempted: 12, Successful: 12, Unsuccessful: 0.
CBR9819I OAM Volume Recovery is ending for volumes CMW008 and N/A.
```

Figure 26. Example of a Normal Completion of Recovery of a Backup Tape Volume

Normal Completion of Recovery of an Optical Volume without All Backup Copies: Figure 27 on page 234 shows recovery of a primary optical volume using the first backup copies of the objects. There are a total of 14 primary objects on the optical volume and its opposite side, but only 12 of the objects have a backup copy. Therefore, only 12 of the objects are successfully recovered.

Note: Because 12 of 12 objects were successfully recovered from the first backup copies of the primary objects, message CBR9863I indicates that zero objects were unsuccessfully recovered. This is because this “Unsuccessful” field represents only the number of objects that the Volume Recovery utility attempted and failed to recover.

```
F OAM,S,RECOVERY,WG310B,BACKUP1
CBR1000I OAM S command execution scheduled.
CBR9800I OAM Volume Recovery starting for volumes WG310B and WG310A.
CBR9827I OAM Volume Recovery.
         The following TAPE volumes are needed for recovery:
         CMW004 *43
CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
         R 43,GO
IEE600I  REPLY TO 43 IS;GO
CBR9852I Volume Recovery Utility processing objects in storage group
         GROUP28 for volume WG310B.
CBR9864I A total of 2 objects on volumes WG310B and WG310A do not have a
         first backup copy.
CBR9103I An error occurred during storage management processing for
         collection GROUP28, object GROUP28.C120.H40. The return code is
         00000008 and the reason code is 00000424.
IEC501A  M 1D19,CMW004,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
CBR9863I Volume Recovery status for volumes WG310B and WG310A. Total:
         14, Attempted: 12, Successful: 12, Unsuccessful: 0.
CBR9819I OAM Volume Recovery is ending for volumes WG310B and WG310A.
```

Figure 27. Example of a Normal Completion of Recovery of an Optical Volume without All Backup Copies

Limited Completion of Recovery of a Backup Volume Due to Error Condition: Figure 28 on page 235 shows the recovery of a backup volume from primary objects that reside on both tape and optical volumes. A total of 24 objects were identified as residing on the backup volume. The OAM Volume Recovery utility attempted to recover the first 17 objects. However, the utility could not process recovery for these 17 objects because a tape drive dynamic allocation failure occurred.

Note: Because the OAM Volume Recovery utility cannot determine if any of the objects were successfully recovered before the failure occurred, the system issues message CBR9862I indicating limited recovery for the volume. The “Successful” and “Unsuccessful” fields in message CBR9863I show statistics of zero.

```
F OAM,S,RECOVERY,CMW004
CBR1000I OAM S command execution scheduled.
CBR9800I OAM Volume Recovery starting for volumes CMW004 and N/A.
CBR9827I OAM Volume Recovery.
         The following TAPE volumes are needed for recovery:
         CMW006
CBR9824I OAM Volume Recovery.
         The following OPTICAL volumes are needed for recovery:
         WG310B WG488A WG488B *42
CBR9810D Reply 'QUIT' to terminate or 'GO' to proceed with recovery.
         R 42,GO
IEE600I  REPLY TO 42 IS;GO
CBR9852I Volume Recovery Utility processing objects in storage group
         GROUP28 for volume CMW004.
IEC501A  M 1D18,CMW006,SL,COMP,OAM,OAM,OAM.PRIMARY.DATA
CBR6425I OAM tape drive dynamic allocation failure for object
         GROUP28.C120.H05 in collection GROUP28 in storage group IMAFIRST on
         tape volume CMW008.
IEF234E  K 1D19,CMW004,PVT,OAM,OAM
IEF234E  K 1D18,CMW006,PVT,OAM,OAM
IEC501A  M 1D18,CMW008,SL,COMP,OAM,OAM,OAM.BACKUP.DATA
CBR9862I Volume Recovery status for volumes CMW004 and N/A is limited.
CBR9863I Volume Recovery status for volumes CMW004 and N/A. Total: 24,
         Attempted: 17, Successful: 0, Unsuccessful: 0.
CBR9819I OAM Volume Recovery is ending for volumes CMW004 and N/A.
```

Figure 28. Example of a Limited Completion of Recovery of a Backup Volume Due to Error Condition

Starting Object Recovery for Single Objects

OAM contains a Single Object Recovery utility for recovering a single object from an optical or tape volume. The system uses a backup copy of the object (either on optical or tape), if any exist. The backup copy that is used for the recovery can reside either on tape or optical media. If more than one backup copy of the object exists, you can select which backup copy of the object will be used for the recovery (first or second backup copy) by specifying BACKUP1 | BACKUP2 on the F OAM,START,OBJRECV command. This causes a new primary copy of the object to be written to the same Object storage group and same media type (optical, tape, or DASD) as the original object. The following are examples of how single object recovery works:

- If the primary object resides on optical disk, the backup copy that is selected for the recovery (on either optical disk or tape) creates a new optical primary copy.
- If the primary object resides on tape, the backup copy that is selected for the recovery (on either optical disk or tape) creates a new tape primary copy.
- If the primary object resides on DASD, the backup copy (on either tape or optical disk) creates a new DASD primary copy.

To recover a single object:

1. Enter the following command:

```
F OAM,START,OBJRECV,collection-name,object-name,BACKUP1|BACKUP2
```

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

3. If the backup volume is an optical volume and does not reside in an optical library, the system issues the following message:

```
CBR4400A Mount volume volser on drive drive-name.  Shelf location is shelfloc.
```

4. If the backup volume is a tape volume, the system issues the following message:

```
IEC501A M drive-Addr,volser,label,,,,data_set_name.
```

5. Mount the optical volume or tape volume that is identified by *volser*.
6. When recovery is complete, the system issues the following message:

```
CBR9830I Single Object Recovery complete for collection collection-name,
         object object-name.
```

The following is a sample of the F OAM,START,OBJRECV command:

```
F OAM,START,OBJRECV,GROUP22,GROUP22.IDVT.H11,BACKUP2

CBR1000I OAM START command execution scheduled.
CBR4401I Volume WG688A mounted on drive P9D1.
CBR4401I Volume WG360A mounted on drive P9D2.
CBR9830I Single Object Recovery complete for
collection GROUP22, object  GROUP22.IDVT.H11.
```

If any errors occur during the single object recovery process, additional messages may be issued identifying the errors For descriptions of these messages, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Starting the Move Volume Utility

OAM provides a Move Volume utility capable of moving objects from a primary or backup source volume (a tape volume or one side of an optical disk) to one or more target volumes for migration purposes.

Note: After the Move Volume utility has successfully completed, the volume expiration date must be updated to ensure that the volume is selected for expiration processing. Use the F OAM,UPDATE,VOLUME command to update the EXPDATE field in the DB2 Volume or Tape Volume tables. See the discussion concerning this command in “Updating Fields in the DB2 Volume Table and the Tape Volume Table” on page 315.

The F OAM,START,MOVEVOL command processing requires that the optical or tape volume specified on the command is either currently managed and controlled by the instance of OAM on which the command was entered (not another OAM within the OAMplex), or if the volume is currently an unmounted shelf-resident volume, it must be available to the instance of OAM on which the command was entered. If the member name associated with the optical or tape volume serial number specified on this command is not blank or is not the member name for this instance of OAM, the command fails. Message CBR1068I is issued.

If the member name associated with the optical or tape volume serial number specified on this command is not the member name for this instance of OAM and is not blanks, this command fails and message CBR1068I is issued.

To move objects from a source volume, enter the following command:

```
F OAM,START,MOVEVOL,volser
```

where *volser* is the volume serial of the source volume from which objects are to be moved.

The system issues the following message:

```
CBR1000I OAM START command execution scheduled.

CBR9850I Move Volume Utility starting for volume volser.

CBR9858I Move Volume Utility status for volume volser.  Total: total,
         Attempted: attempted, Successful: successful,
         Unsuccessful: unsuccessful.

CBR9859I Move Volume Utility ending for volume volser.
```

If any errors occur or contention exists due to concurrent processing, you must take appropriate actions to correct the errors or minimize contention. You may then re-invoke the Move Volume utility to continue movement of objects from the source volume. See "Statistics Provided by the Move Volume Utility" on page 238 for examples of various conditions that can exist and the resulting statistics provided by the utility.

Notes:
1. For a given optical disk (consisting of two volumes), only one Move Volume utility (which moves data from one of the two volumes on the disk) or one Volume Recovery utility (which recovers data on both volumes on the disk) can be started for a disk, but not both.
2. For a give tape volume, only one Move Volume utility or one Volume Recovery utility can be started for the volume, but not both
3. You cannot start the Move Volume utility concurrently for both volumes on an optical disk.
4. You cannot start the Move Volume utility for a volume that is on either side of an optical disk that is currently being recovered by the Volume Recovery utility.
5. You can start one or more Move Volume utilities only after determining that there are sufficient resources available to the utility. Consider the resources required by other OSMC functions, such as the OSMC storage management cycle, which may run concurrently with the Move Volume utility.
6. To avoid contention, it is recommended that you do not start the Move Volume utility for multiple volumes that can cause concurrent references to objects in the same Object storage group. References to objects in the same Object storage group concurrently can occur when there are two primary source volumes in the same Object storage group, two backup source volumes where

the primary volumes containing the objects are in the same Object storage group, or any combination of the above. Figure 29 illustrates these relationships.

7. If the member name associated with the optical or tape volser specified on the MOVEVOL command is not the member name for this instance of OAM or not blank, the command fails and the message CBR1068I is issued.

In this example there are four Object storage groups, two primary volumes (A and B), and two backup volumes (C and D) as shown in Figure 29.

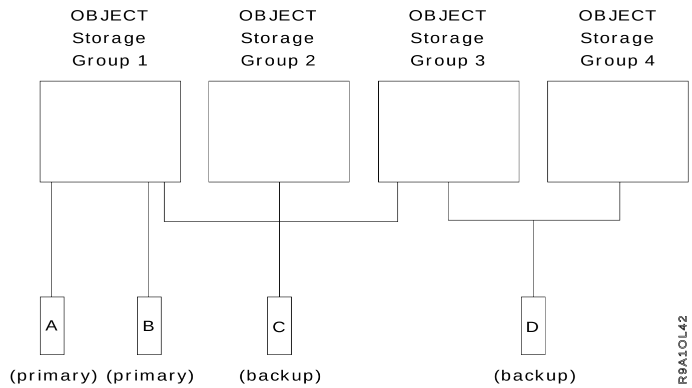

Figure 29. Object Storage Group and Volume Relationship

The Move Volume utility *should not* be started concurrently for the following situations:

- A and B—Two primary source volumes in the same Object storage group. This would cause access to objects in Object storage group 1 concurrently during the two Move Volume utilities processing primary source volumes A and B.
- A and C or B and C—A primary source volume and a backup source volume where the primary volumes containing the objects are in the same Object storage group. This would cause access to objects in Object storage group 1 concurrently during the two Move Volume utilities processing primary source volume A and backup source volume C.
- C and D—Two backup source volumes where the primary volumes containing the objects are in the same Object storage group. This would cause access to objects in Object storage group 3 concurrently during the two Move Volume utilities processing backup source volumes C and D.

The Move Volume utility *can* be started concurrently for the following situations:

- A and D or B and D—A primary source volume and a backup source volume where the primary volumes containing the objects are in different Object storage groups. The Move Volume utility for primary source volume A would only cause access to objects in Object storage group 1, and the Move Volume utility for backup source volume D would only cause access to objects in Object storage groups 3 and 4.

Statistics Provided by the Move Volume Utility

The following examples illustrate the messages that are provided by the Move Volume utility to provide statistics for a variety of conditions.

Normal Completion: Figure 30 on page 239 shows an example of the messages provided in the case of a normal completion of the Move Volume utility.

```
F OAM,START,MOVEVOL,WG360A
CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume Utility starting for volume WG360A.
CBR9852I Move Volume Utility processing objects in storage group GROUP22
 for volume WG360A.
CBR4401I Volume WG360B mounted on drive P9D2.
CBR9858I Move Volume Utility status for volume WG360A. Total: 5,
Attempted: 5, Successful: 5, Unsuccessful: 0.
CBR9859I Move Volume Utility ending for volume WG360A.
```

Figure 30. Example of Messages Returned after a Normal Completion of MOVEVOL

In this example, five objects were identified on the backup source volume and all five objects were successfully moved.

Notes:

1. 1 to *n* CBR9852I messages are issued (one for each Object storage group processed).
2. “Successful” and “Unsuccessful” counts always add up to the “Attempted” count on a normal completion.

Normal Completion with Contending System Activity: Figure 31 shows an example of the messages provided for a normal completion of the Move Volume utility where the utility has detected that there is other system activity contending with its processing.

```
F OAM,S,MOVEVOL,BACK01
CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume utility starting for volume
         BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP00 for volume BACK01.
CBR9131I BACK01 CBRHDUPD attempted to update collection
         GROUP00.B, object GROUP00.AAAA.A.STO5.OBJ1 in
         storage group GROUP00. The directory entry
         for the object was already changed.
CBR9131I BACK01 CBRHDUPD attempted to update collection
         GROUP00.B, object GROUP00.AAAA.A.STO5.OBJ2 in
         storage group GROUP00. The directory entry
         for the object was already changed.
CBR9858I Move Volume utility status for volume BACK01.
         Total: 10, Attempted: 10, Successful: 8,
         Unsuccessful: 2.
CBR9859I Move Volume utility ending for volume BACK01.
```

Figure 31. Example of Messages Returned after a Normal Completion with Contention

In this example, 10 objects are identified on the backup source volume and all 10 objects were attempted to be moved; however, only 8 of the 10 attempted objects have been successfully moved. The remaining 2 of the 10 attempted objects have not been successfully moved due to contention with other system activity for those objects.

Limited Completion: Figure 32 on page 240 shows an example of the messages provided for “limited” completion of the Move Volume utility where the utility processing was not complete due to contention, errors, and so on.

```
F OAM,S,MOVEVOL,BACK01
CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume utility starting for volume
         BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP00 for volume BACK01.
CBR9852I Move Volume utility processing objects in
         storage group GROUP01 for volume BACK01.
CBR9855I Move Volume utility processing limited for
         volume BACK01.
         Less objects than expected were found in
         collection GROUP01.A.
CBR9857I Move Volume utility status for volume BACK01
         is limited.
CBR9858I Move Volume utility status for volume BACK01.
         Total: 2, Attempted: 1, Successful: 0,
         Unsuccessful: 0.
CBR9859I Move Volume utility ending for volume BACK01.
```

Figure 32. Example of Messages Returned after a Limited Completion

In this example, 2 objects are identified on the backup source volume and there has been an attempt to move only 1 object. Because the utility cannot successfully complete due to a discrepancy in the number of objects in a collection being processed, it is not possible to determine whether the 1 object that was attempted to be moved was successfully moved or not.

Note: "Successful" and "Unsuccessful" counts are always zero when message CBR9857I indicates that the status is "limited".

Not Available Completion: Figure 33 shows an example of the messages provided for "not available" completion of the Move Volume utility where the utility processing is not complete due to contention, errors, and so on, and has not been able to determine the number of objects to be moved.

```
F OAM,S,MOVEVOL,BACK01
CBR1000I OAM S command execution scheduled.
CBR9850I Move Volume utility starting for volume
         BACK01.
CBR9089I No storage groups defined in the active
         configuration.
CBR9856I Move Volume utility stopping for volume
         BACK01.
CBR9857I Move Volume utility status for volume BACK01
         is not available.
CBR9859I Move Volume utility ending for volume BACK01.
```

Figure 33. Example of Messages Returned after a Not Available Completion

In this example, the utility is not even able to determine how many objects needed to be moved due to an error condition.

Note: Message CBR9858I is *not* issued when message CBR9857I indicates that the status is "not available" since there are no statistics that can be reported.

Status of the Source Volume: During the execution of the Move Volume utility, the source volume is made ineligible for writing.

At the completion of the utility, the WRITABLE status of the source volume is restored to its previous state prior to the execution of the utility. In addition, every object that was moved is deleted from the source volume. This includes logical deletion whereby OAM no longer maintains directory information on the former location of the object, as well as actual physical deletion in the case of rewritable media to reclaim the space once occupied by the object.

When all objects are moved from both sides of an optical WORM source volume, you may want to perform the following tasks:

- Update the expiration date of the volume using the F OAM,UPDATE,VOLUME command so that OSMC will select and validate expiration handling of the volume.
- EJECT the volume if it is resident in a library
- Dispose of the volume, after OSMC has validated the expiration, in accordance with your installation policies and follow the manufacturer's recommendations if the volume is to be destroyed.

  Caution: Optical disk cartridges may contain materials that are regulated for disposal. Dispose in accordance with local, state, and federal regulations.

Status of the Objects Following the Completion of the Utility: At the conclusion of the utility, the OAM DB2 object directory table row for each object that was moved is updated to reflect the new object location. Additionally, each object is scheduled for processing during the next OSMC storage management cycle.

The next OSMC storage management cycle selects each object for processing, calculates the expiration date for the object, and updates the expiration date for the volumes on which the objects are written. As a result, you should expect an increase in the number of objects processed during the OSMC storage management cycle following the movement of objects.

Starting Automatic Access to Backup Copies of Objects

OAM can obtain a backup copy of an object if the primary copy of the object is resident on a removable media volume that is unavailable for any of the following reasons:

- Marked unreadable (possibly due to damage or destruction)—**UNREAD**
- Resides in a library that is currently or pending offline—**OFFLINE**
- Resides in a library that is currently marked nonoperational—**NOTOPER**
- Not available for any of the above reasons (nonspecific)—**ALL**

Automatic access to backup copies of objects must be active for one or all of the specific reasons before OAM attempts to obtain the backup copy. For more information on this function, see "Accessing Backup Objects Automatically" on page 193.

With the use of the access backup function, it is unnecessary for the application to specify the VIEW=BACKUP1 or VIEW=BACKUP2 parameter to obtain the backup copy of the object.

Note: This access to backup is limited to retrieval requests that are issued by the OSREQ macro. OSMC does not support access to backup for moving objects from removable media to the DB2 tables.

The following actions can be taken to ensure the retrieval of backup copies of objects. These actions can be performed any time, even when access backup is already active:

- Make a physical inventory of the damaged removable media, noting the volumes residing on the unusable media.
- Repair or replace the hardware as required.
- Obtain all backup removable media volumes containing objects that are related to the objects on the damaged removable media.
- Restore all the OAM DB2 databases from the latest image copy and complete forward recovery through the most recent updates that are available.
- Use ISMF to make any necessary updates to the optical configuration database for any alterations or updates that are needed for your hardware configuration.
- Use the F OAM,UPDATE,VOLUME,*volser* command to alter the optical configuration database Volume Table or Tape Volume Table, changing the READABLE column value to "N" (NO) for each pair of volumes that is identified as residing on damaged (unreadable) media.

Performing one or more of these steps allows normal application processing to occur until object recovery operations restore the primary copies of objects.

The following MVS command syntax starts the automatic access to backup copies function:

```
                                                ┌─BACKUP1─┐
►►──┬─MODIFY─┬──OAM,──┬─START─┬─,─AB─,─reason─,─┤         ├──────────────►◄
    └─F──────┘        └─S─────┘                 └─BACKUP2─┘
```

To start the automatic access to backup copies function processing:

1. Enter the following command:

```
F OAM,START,AB,reason,BACKUP1 | BACKUP2
```

The following are valid values and descriptions for the *reason* keyword:

UNREAD
: This is the default. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is marked not readable, the backup copy of the object is retrieved.

OFFLINE
: When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is in a library that is currently offline or pending offline, the backup copy of the object is retrieved.

NOTOPER
: When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is in a library that is currently marked nonoperational, the backup copy of the object is retrieved.

ALL
: When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is not available for any of the above reasons, the backup copy of the object is retrieved.

The following are the valid, optional options for the VIEW parameter:

BACKUP1 | BACKUP2
When a primary copy of an object that is to be read by an application is not available for the specified reason, and BACKUP1 is specified on the START command for automatic access to backup, an attempt is made to retrieve the object from the first backup copy of the object. If BACKUP2 is specified, an attempt is made to retrieve the object from the second backup copy of the object. BACKUP1 is the default if neither BACKUP1 nor BACKUP2 is specified.

2. The system issues the following message:

```
CBR1000I OAM START command execution scheduled.
```

3. Once the access backup processing starts, the system issues the following message:

```
CBR1090I OAM Access Backup processing started for reason using the
         {first | second} backup copy.
```

4. Access backup processing remains active until a STOP,AB,*reason* command is issued. Then the system issues the following message:

```
CBR1091I OAM Access Backup processing stopped for reason.
```

5. If access backup processing is currently active and the operator tries to issue another START,AB,*reason* command, the system issues the following message:

```
CBR1092I OAM Access Backup processing already started for reason.
```

The following is a sample of using the F OAM,START,AB command to retrieve the second backup copy of an object:

```
F OAM,START,AB,UNREAD,BACKUP2

CBR1000I OAM START command execution scheduled.
CBR1090I OAM Access Backup processing started for UNREADABLE VOLUMES using the
             second backup copy.
```

The following is a sample of using the F OAM,START,AB command to retrieve the first backup copy of an object:

```
F OAM,START,AB,UNREAD,BACKUP1

CBR1000I OAM START command execution scheduled.
CBR1090I OAM Access Backup processing started for UNREADABLE VOLUMES using the
         first backup copy.
```

Note: Stopping OAM and starting OAM does not affect the status of automatic access to backup. If AB has been started and OAM is stopped, AB is active when OAM is started again.

Varying Optical Drives and Libraries

You can vary optical drives and optical libraries online or offline. This means that you can control whether the system can access the optical drive or optical library.

The vary online command requires that all associated libraries and drives for the library or drive being brought online be currently managed and controlled by (online to) the OAM targeted for the vary request, or not currently managed and controlled by any OAM in an OAMplex (offline to all systems), thereby making it eligible to be brought online to the target OAM. In other words, if a library or drive is to be brought online to an instance of OAM, no associated libraries or drives in the same 3995 subsystem can be online to any other instance of OAM in the OAMplex.

The vary offline command requires that the library or drive being taken offline be currently managed and controlled by the OAM targeted for the vary request.

Note: Changing the offline status of the optical library does not affect the online/offline status of the library-resident optical drives contained within the library. Use the VARY SMS command to vary an optical drive online or offline, or an optical library online or offline.

Varying an Optical Drive Online or Offline

Before OAM can allow a drive to be brought online to an instance of OAM, 3995 drive processing must verify that the drive is not online to another instance of OAM in the OAMplex, or that the OAM that currently manages and controls the drive has terminated or does not have a valid XCF member defined.

Drive vary processing must check the status of the optical library to which this drive is physically attached. If this drive is to be brought online, the library where it resides must not be online to another OAM in the OAMplex. It also checks the status of all the other libraries associated with the drive and the drives associated with those libraries.

Drive vary processing must also check the status of other drives in the optical library to which this drive is attached. If this drive is to be brought online, no other drives in the library where the drive resides can be online to another instance of OAM in the OAMplex.

SMS validates the specified system ID that is targeted for the vary request; it also verifies that the specified drive is defined as connected to the target system ID in the ACDS.

The following SMS command syntax varies optical drives online and offline:

```
                                          ┌─*─────────┐
►►──┬─VARY─┬──SMS,──┬─DRIVE─┬──(drive-name,──┼─system_id─┼──)─,──┬─┬─ONLINE─┬───┬──────►◄
    └─V────┘        └─DRI───┘                 └─*─────────┘       │ └─ON─────┘   │
                                                                  └─┬─OFFLINE─┬─┘
                                                                    └─OFF─────┘
```

DRIVE(*drive-name***)**
Specifies the name of the optical drive to be varied online or offline. If the name is not specified or the specified drive is not defined in the SMS configuration, an error message is displayed.

***** Indicates that the target of the drive vary request is the current OAM.

system-id
Specifies the MVS system that is the target of the vary request.

Note: For optical drives, only one system ID can be specified.

ONLINE
Specifies that the optical drive is to be varied online.

OFFLINE
Specifies that the optical drive is to be varied offline.

Here is an example of the command to vary an optical drive online:

```
VARY SMS,DRIVE(drive-name,system-id),ONLINE
```

Here is an example of the command to vary an optical drive offline:

```
VARY SMS,DRIVE(drive-name,system-id),OFFLINE
```

Note: You can demount an optical disk cartridge on a stand-alone or operator-accessible drive by varying the drive offline.

Varying a Real Optical Library Online or Offline

Before OAM can allow a library to be brought online to an instance of OAM, 3995 library processing verifies that the 3995 optical library is not online to another instance of OAM in the OAMplex, or that the OAM that currently owns the library is failed (the XCF member name is not defined or is no longer valid for the OAMplex).

Library vary processing must also check drive status of the optical drives that are physically attached to this library and make certain that the drives are not online to another instance of OAM, or that the OAM that currently manages and controls the drive has terminated or does not have a valid XCF member defined. It also checks the status of associated libraries and controller and the status of the drives in those libraries.

SMS validates the specified system ID targeted for the vary request, and verifies that the specified drive is defined as connected to the target system ID in the ACDS.

The following SMS command syntax varies optical libraries online and offline:

```
►►──┬─VARY─┬──────────────────────────────────────────────────────────────►
    └─V────┘

                                      ┌─*─────────┐
►─SMS,──┬─LIBRARY─┬──(library-name,──┼─system_id─┼──)─,──┬─┬─ONLINE──┬─┬──────────►◄
        └─LIB─────┘                   └─*─────────┘       │ └─ON──────┘ │
                                                          └─┬─OFFLINE─┬─┘
                                                            └─OFF─────┘
```

LIBRARY(*library-name***)**
Specifies the name of the real optical library that is to be varied online or offline. If the name is not specified or the specified library is not defined in the SMS configuration, an error message is displayed.

Note: Pseudo libraries cannot be varied online or offline. If a pseudo library name is used in attempting to vary a library online or offline, an error message is displayed.

system-id
Specifies the MVS system that is the target of the vary request.

Note: For optical libraries, only one system ID can be specified.

***** Indicates that the target for the library vary request is the current OAM system.

ONLINE
Specifies that the optical library is to be varied online.

OFFLINE
Specifies that the optical library is to be varied offline.

Note: In order to change the system where an optical library is currently online, the library and drives must be first varied offline to the system where they are currently online. Then, after successfully bringing the 3995 device addresses online to MVS, the optical library and drives can be varied online to OAM.

Here is an example of the command to vary an optical library online:

```
VARY SMS,LIBRARY(library-name,system-id),ONLINE
```

Here is an example of the command to vary an optical library offline:

```
VARY SMS,LIBRARY(library-name,system-id),OFFLINE
```

Varying an optical library offline does not instantaneously cause the library to be varied offline if there is activity currently in process or on the queue to be processed. The vary command activates immediately but does not complete until after the activity on the drives has completed. Once the library is pending offline, all library specific activity to that library fails. Write activity to a storage group that spans multiple libraries including the offline library is scheduled to one of the other libraries.

Entering an Optical Disk into an Optical Library

When OAM needs an additional optical disk volume to satisfy an out-of-space condition for a particular storage group which contains a real library, the following messages are displayed:

- For 9246 stand-alone libraries:

```
CBR2211E Enter an optical disk into library library-name to relieve the out
         of space condition in storage group storage-group-name.
```

- For 3995 libraries:

```
CBR2213I No space left in storage group storage-group-name.

CBR2217E Enter an optical disk cartridge that is compatible with the DEFAULT
         MEDIA TYPE library-default-media-type into library library-name
         to relieve the out of space condition in storage group
         storage-group-name.

CBR2550I Optical disk entry into library library-name scheduled.

CBR4401I Volume volser-1 mounted on drive drive-name.

CBR2100I Volumes volser-1 and volser-2 entered into library.
```

You can enter the following in response to message CBR2211E or CBR2217E:
- An unlabeled optical disk
- A scratch optical disk
- An optical disk whose volumes are assigned to the requested storage group, writable, not full, and not write-protected

If all slots in the optical library are either occupied by an optical disk cartridge or reserved for an optical disk cartridge mounted on one of the library optical drives, the optical library is considered to be full. This condition is detected by displaying the optical library status and checking the CBR1110I message field EMP SLT (empty slot) for a value of zero (see “Displaying Library Detail Status” on page 290). The system automatically runs library space management for this library before requesting an additional optical disk cartridge.

Note: When entering an optical disk cartridge into a library with 3995 optical disk drives, the media type of the cartridge must be consistent with the default media type for the library. If the media type of the cartridge is not consistent with the default media type, the cartridge is ejected. If the media is being isolated for specific applications, assign the cartridges to a storage group. The cartridges cannot be entered as scratch volumes.

If the media type entered is compatible with the library default media type but is not write compatible with the drives in the library, the out-of-space condition is not relieved. An example of this scenario might be:

- A 3995-Cxx library containing 3995-SW3 drives
- With a library default media type of 3995
- And the volume entered is a single-density WORM cartridge

In this case, the out-of-space condition will still exist because the 3995-SW3 drives cannot write to a single-density WORM cartridge.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

Note: Because WORM optical volumes that are full or have very little free space are not useful as scratch volumes, the operator is notified, via a message, if the kilobytes free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. This gives the operator the opportunity to fail the cartridge entry process, causing the cartridge to be ejected from the library.

Entering an Unlabeled Optical Disk into a 9246 Optical Library

To enter an unlabeled optical disk into a 9246 optical library, do the following:

1. Put the optical disk cartridge into the optical library input/output station. The system issues the following message:

```
CBR2550I Optical disk entry into library library-name scheduled.
```

2. The optical disk moves from the input/output station when an optical drive becomes available. The system issues the following message and waits for a reply:

```
CBR4412D Enter VOLSER for volume on drive drive-name in library library-name.
```

3. Provide a 1 to 6 character unique volume serial number. The system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

4. Provide 1 to 64 characters of owner identification. The system issues the following message:

```
CBR4415I Volume label written to volume on drive drive-name.  Volume serial
         number is volser-1.
```

   In the message text, *volser-1* is replaced by the volume serial number of the first side of the optical disk, and *drive-name* is replaced by the name of the optical drive in the optical library where the volume to be labeled is currently mounted.
5. The optical disk is automatically flipped over and the procedure begins again for labeling the other side of the optical disk. The system issues the following message and waits for a reply:

```
CBR4412D Enter VOLSER for volume on drive drive-name in library library-name.
```

6. Provide a 1 to 6 character unique volume serial number. The system issues the following message:

```
CBR4415I Volume label written to volume on drive drive-name.  Volume serial
        number is volser-2.
```

7. In the message text, *volser-2* is replaced by the volume serial number of the second side of the optical disk, and *drive-name* is replaced by the name of the optical drive in the optical library where the volume to be labeled is currently mounted. The system issues the following message:

```
CBR4432D Enter storage-group-name for volumes volser1 and volser2, or reply 'U'
        to assign to scratch.
```

8. If the volumes are to be assigned to scratch status, reply ‘U’ to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following messages:

```
CBR4401I Volume volser-2 mounted on drive drive-name.

CBR2100I Volumes volser-1 and volser-2 entered into library: library-name.
```

   The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser-1* and *volser-2* are replaced by the volume serial numbers you entered for the optical volumes.

Entering an Unlabeled Optical Disk into a 3995 Optical Library

To enter an unlabeled optical disk into a 3995 optical library:

1. Put the optical disk cartridge into the optical library input/output station. The system issues the following message:

```
CBR2550I Optical disk entry into library library-name scheduled.
```

2. The optical disk moves from the input/output station when an optical drive becomes available. The system issues the following message for each side of the optical disk cartridge.

```
CBR3381I Volume mounted on drive on drive-name contains an unrecognized format.
```

3. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive-name has unrecognized media format.
       Reply 'F' to format or 'C' to cancel.
```

Note: This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of ‘F’ causes the volume to be formatted and any data currently existing on the volume is destroyed. Replying ‘C’ at this point causes the formatting processing to stop, leaving only the one side of the optical cartridge formatted. Further information about commands from the 3995 dynamic console can be found in the *3995 Operator’s Guide*.

Attention: Formatting a rewritable optical disk volume is a time consuming process. Do not interrupt this process by assuming that OAM is inactive during this time frame. OAM issues a completion message when this process is finished.

4. To continue the label process, reply with an ‘F’. To cancel the label process, reply with an ‘C’. If a reply of ‘F’ is entered, the system issues the following message and waits for a reply:

```
CBR4412D Enter VOLSER for volume on drive drive-name in library library-name.
```

5. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser.  Reply 'U'
       to use this volser, or 'R' to retry.
```

6. The system issues the following message and waits for a reply:

```
CBR4439D Enter VOLSER for opposite side of volume volser in drive drive-name.
```

7. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser.  Reply 'U'
       to use this volser, or 'R' to retry.
```

8. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

9. Provide 1 to 64 characters of owner identification. The system issues the following message:

```
CBR4432D Enter storage-group-name for volumes volser-1 and volser-2, or reply 'U'
         to assign to scratch.
```

10. If the volumes are to be assigned to scratch status, reply ‘U’ to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following message:

```
CBR4401I Volume volser-2 mounted on drive drive-name.
```

11. The system issues the following message:

```
CBR2100I Volumes volser-1 and volser-2 entered into library: library-name.
```

The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser-1* and *volser-2* are replaced by the volume serial numbers you entered for the optical volumes.

Entering a Labeled Optical Disk into an Optical Library

To put a labeled optical disk into the optical library without prompting from the system:

1. Put the optical disk cartridge into the optical library input/output station. The system issues the following message:

```
CBR2550I Optical disk entry into library library-name scheduled.
```

Note: When entering an optical disk cartridge into a library with 3995 optical disk drives, the media type of the cartridge must be consistent with the default media type for the library and must be compatible with the library resident drives. If the media type of the cartridge is not consistent with the default media type or compatible with the library resident drives, the cartridge is ejected. If the media is being isolated for specific applications, assign the cartridges to a storage group. The cartridges cannot be entered as scratch volumes.

2. The optical disk moves from the input/output station when an optical drive becomes available. The system reads and verifies the volume label on each side of the optical disk. The system issues the following messages:

```
CBR4420I Volume table did not contain information for volser-1 on drive
         drive-name.

CBR4420I Volume table did not contain information for volser-2 on drive
         drive-name.

CBR4432D Enter storage group name for volumes volser-1 and volser-2,
         or reply 'U' to assign to scratch.

CBR4401I Volume volser-1 mounted on drive drive-name.

CBR2100I Volumes volser-1 and volser-2 entered into library library-name.
```

Ejecting an Optical Disk

Use the MVS LIBRARY EJECT command to eject a specific optical disk from an optical library. This command requires that the optical volume specified is currently managed and controlled by the instance of OAM on which the command was entered. If the member name associated with the volume serial number is not the member name for this instance of OAM, the command fails and message CBR1068I is issued.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, then that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

The following command syntax ejects a specific optical disk:

```
►►──┬─LIBRARY─┬──┬─EJECT─┬─,volser──┬──────────────┬──┬─────────────┬────────────►◄
    └─LI──────┘  └─E─────┘           ├─,──LOCATION─┤  ├─,──LIBRARY─┤
                                     └─,──L────────┘  └─,──LIB─────┘
```

The following describes the values of the options for the EJECT command:

volser
: Specifies the volume serial number of one of the two optical volumes on the optical disk that is to be ejected.

LOCATION
: Prompts the operator to specify the new shelf location information for the ejected optical disk volumes.

LIBRARY
: Prompts the operator to specify a pseudo library name to which the ejected volume will be assigned.

To eject an optical disk from an optical library, enter the following command:

```
LI EJECT,volser,LOCATION,LIBRARY
```

An optical disk can arrive in the input/output station without an operator request in the following ways:

- The system can eject the optical disk as a result of the Library Space Manager utility running on that library.

  The system ejects an optical disk volume if there are no scratch optical volumes in the optical library, no empty slots in the optical library, and the system is attempting to write to the storage group that has an out-of-space condition.
- The system ejects an optical disk volume if the volume is expired.
- The system ejects an optical disk volume if an unknown volume results in an error during volume entry sequence.
- A storage administrator can specify an EJECT line operator next to an optical volume serial number on an ISMF Mountable Optical Volume List panel.
- The system responds to a REMAP command against the library control inventory.

Note: Demount of an optical disk on an operator-accessible or stand-alone drive can be accomplished by varying the drive offline.

Specifying the Shelf Location

If a shelf location has not been specified previously for the optical disk being ejected or a new shelf location was requested on EJECT command, the system issues the following messages and waits for the appropriate responses:

```
CBR1000I OAM EJECT command execution scheduled.
```

```
CBR2600A Specify shelf location for volumes volser-1 and volser-2.
```

Provide 1 to 32 characters of shelf information. The system issues the following messages:

```
CBR2603A Specify pseudo library name for volume volser-1.
```

Provide the name of the associated pseudo library for this volume. The system issues the following messages:

```
CBR3001A Remove cartridge from I/O station on library library-name.  Place in
         shelf location shelf-location.
```

```
CBR3122I Volumes volser-1 and volser-2 were ejected from library library-name,
         shelf location is shelf-location.
```

Associating Pseudo Libraries

When an optical volume is ejected from a library, it must be assigned to a pseudo library. In previous releases, shelf resident volumes were associated with a pseudo optical library that represented volumes of that media type. Now, with no device type association for pseudo libraries, the assignment of a shelf resident volume to a pseudo library must be done when the volume is ejected from a library. For more information regarding pseudo libraries, see "Pseudo Optical Library Concept" on page 34.

If the volume being ejected already has a pseudo library name associated with it in its volume record, that pseudo library continues to be used unless it is overridden with the LIBRARY keyword on the EJECT command. If the volume being ejected does not already have a valid pseudo library associated with it, the default pseudo library associated with the library from which the volume was ejected may be used, if one exists, unless it is overridden with the LIBRARY keyword on the EJECT command. If the optical volume being ejected does not already have a valid pseudo library associated with it, or the library from which the volume was ejected does not have a default pseudo library in its definition, or the LIBRARY keyword was specified on the EJECT command, the system issues message CBR2602A or CBR2603A, or both.

```
CBR2602A Eject pending for volser in r-library-name. Default pseudo library
         is p-library-name. Reply 'U' to use, or 'R' to respecify.
```

```
CBR2603A Specify pseudo library name for volume volser.
```

If the volume being ejected has an invalid pseudo library associated with its volume record and the library from which the volume is being ejected has a default pseudo library in its SCDS definition, message CBR2602A is issued. This message asks if the default is to be used or if another pseudo library name is requested.

If the library from which the volume is being ejected does not have a default pseudo library in its SCDS definition, or "R" was replied to message CBR2602A, then message CBR2603A is issued. This message requests a pseudo library name to assign the volume to when it becomes shelf resident.

If a pseudo library name specified in response to message CBR2603A is not valid in the current ACDS, or the volume record has an invalid pseudo library name associated with it, message CBR2604I is issued, followed by message CBR2603A.

```
CBR2604I Volume volser cannot be assigned to pseudo library p-library-name, it is
         not a valid pseudo library definition in the active SMS configuration.
```

```
CBR2603A Specify pseudo library name for volume volser.
```

Removing the Optical Disk Cartridge

When you receive the following message, remove the optical disk cartridge from the input/output station of the specified optical library and return it to the shelf location indicated:

```
CBR3001A Remove cartridge from I/O station on library library-name.  Place in
         shelf location shelfloc.
```

Attention: Upon ejection of a cartridge, immediately remove it from the input/output station; otherwise, performance is degraded.

Mounting an Optical Disk on a Stand-Alone or Operator-Accessible Drive

When the system requests a shelf-resident volume, the following message is displayed requesting you to mount an optical disk cartridge on a stand-alone or operator-accessible optical drive:

```
CBR4400A Mount volume volser on drive drive-name.  Shelf location is shelfloc.
```

If the *volser* is ??????, the request is for an unlabeled optical volume. Locate an unlabeled optical disk cartridge and load it on drive *drive-name*. See “Labeling an Optical Disk on a 9247 Stand-Alone Drive” to label a disk on a stand-alone drive and “Labeling an Optical Disk on a 3995 Operator-Accessible Drive” on page 259 to label a disk on an operator-accessible drive.

If a volume serial number is specified in *volser*, the request is for an already labeled optical volume. Locate the optical disk cartridge and load it into the drive *drive-name*. The system issues the following message:

```
CBR4401I Volume volser mounted on drive drive-name.
```

The volume is now ready for the system to use.

Notes:

1. You can also mount an optical disk on a stand-alone or operator-accessible optical drive without waiting for a request from the system. To mount the optical disk, vary the stand-alone or operator-accessible optical drive offline, load the optical disk cartridge into the drive, and then vary the drive online.
2. For efficiency purposes, the system places a response time limitation of five minutes from the time the operator mount message is received on the console to the time the mount is completed. If the mount is not completed within the allotted time, the operator has the option of canceling or retrying the optical disk mount.

Demounting and Removing an Optical Disk Cartridge from a Stand-Alone or Operator-Accessible Drive

To demount an optical disk cartridge from a stand-alone or an operator-accessible drive, the drive must be varied offline using the OAM VARY command. This causes the cartridge to be demounted so that the operator may then remove the cartridge.

Labeling an Optical Disk on a 9247 Stand-Alone Drive

When there are no scratch optical volumes on the shelf, and a storage group which spans a pseudo library is out of space, the following message is displayed requesting you to label an unlabeled optical disk on a stand-alone optical drive:

```
CBR2212E Use the OAM LABEL command to label optical disks for shelf use
         to relieve the out of space condition in storage group
         storage-group-name.
```

Note: You can also use a library optical drive to label an optical disk. See “Entering an Optical Disk into an Optical Library” on page 247 to label an optical disk in an optical library.

Use the OAM LABEL command to label optical volumes on stand-alone optical drives. OAM rejects the LABEL command if there is no stand-alone optical drive online and operational.

The following syntax labels an optical disk on a stand-alone drive:

```
                                       ┌─9247───────┐
►►──┬─MODIFY─┬──OAM,──┬─LABEL─┬──,──┴─drive_name─┴──,─pseudo-library──────────────►◄
    └─F──────┘        └─L─────┘
```

The following describes the values of the options for the LABEL command:

drive_name
: Refers to a specific IBM 9247 stand-alone drive on which the label command is processed. The optical media being labeled must be compatible with the drive requested for reading and writing purposes. The name of the optical drive must be defined in the active SCDS configuration.

pseudo-library
: Refers to the name of the pseudo library whose drives are to be considered for the label request, and the pseudo library with which the newly labeled volume will be associated. If this keyword is not specified, the default will be the pseudo library associated with the stand-alone drive that is performing the label function. If a pseudo library name specified on the LABEL command is invalid for the current ACDS, message CBR1305I is issued and the command fails.

If a pseudo-library name is specified on the LABEL command, the device type of 9247 must be specified because pseudo libraries no longer require a device type affinity, so that mixed devices and media types may be associated with a pseudo-library. If the drive name is specified, specification of a pseudo-library is ignored.

To write the label on two volumes of an optical disk on a stand-alone drive:

1. Enter the following command:

```
F OAM,LABEL,9247
```

2. The system issues the following message:

```
CBR1000I OAM LABEL command execution scheduled.
```

3. When a stand-alone optical drive becomes available, the system issues the following message and waits for you to mount an optical disk cartridge.

```
CBR4400A Mount volume ?????? on drive drive-name.  Shelf location is ??????.
```

4. Put an unlabeled optical disk cartridge into the stand-alone optical drive. The system issues the following message and waits for a reply:

```
CBR4405D Enter VOLSER for volume on drive drive-name.
```

5. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the

volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser.
         Reply 'U' to use this volser, or 'R' to retry.
```

6. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

7. Provide 1 to 64 characters of owner identification. The system issues the following message and waits for a reply:

```
CBR4423D Enter shelf information for volume volser on drive drive-name.
```

8. Provide 1 to 32 characters of shelf information. The system issues the following message:

```
CBR4415I Volume label written to volume on drive drive-name.  Volume serial
         number is volser.
```

9. In the message text *volser* is replaced by the volume serial number of the first side of the optical disk, and *drive-name* is replaced by the name of the stand-alone optical drive where you mounted the optical disk. The system issues the following message and waits for a reply:

```
CBR4430A Remove and flip cartridge on drive drive-name.
```

10. Remove the optical disk cartridge, flip it over, and reinsert it into the stand-alone optical drive. The system issues the following message and waits for a reply:

```
CBR4405D Enter VOLSER for volume on drive drive-name.
```

11. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D VOLSER for unlabeled volume in drive drive-name is volser.
         Reply 'U' to use this volser, or 'R' to retry.
```

12. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4415I Volume label written to volume on drive drive-name.  Volume
         serial number is volser.
```

13. In the message text, *volser* is replaced by the volume serial number of the second side of the optical disk and *drive-name* is replaced by the name of the stand-alone optical drive where you mounted the optical disk. The system issues the following message:

```
CBR4432D Enter storage-group-name for volumes volser1 and volser2,
         or reply 'U' to assign to scratch.
```

14. If the volumes are to be assigned to scratch status, reply ‘U’ to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following messages:

```
CBR4401I Volume volser-2 mounted on drive drive-name.

CBR2102I LABEL function complete for volumes volser-1 and volser-2.
```

The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser-1* and *volser-2* are replaced by the volume serial numbers you entered for the optical volumes.

Labeling an Optical Disk on a 3995 Operator-Accessible Drive

When there are no scratch optical volumes on the shelf, and a storage group which contains a pseudo library is out of space, the following messages are displayed requesting you to label an unlabeled optical disk on the operator-accessible optical drive:

```
CBR2213I No space left in storage group storage-group-name.

CBR2212E Use the OAM LABEL command to label optical disks for shelf use
         to relieve the out of space condition in the storage group
         storage-group-name.


CBR1000I OAM L command execution scheduled.

CBR4400A Mount volume ?????? on drive drive-name.  Shelf location is ??????

CBR4419I Previously labeled volume volser-1 was mounted on drive drive-name.

CBR4423D Enter shelf information for volume volser-1 on drive drive-name.


CBR4430A Remove and flip cartridge on drive drive-name.

CBR4419I Previously labeled volume volser-2 was mounted on drive drive-name.

CBR4432D Enter storage group name for volumes volser-1 and volser-2,
         or reply 'U' to assign to scratch.

CBR4401I Volume volser-2 mounted on drive drive-name.

CBR2102I LABEL function complete for volumes volser-1 and volser-2.
```

Note: You can also use a library optical drive to label an optical disk. See “Entering an Optical Disk into an Optical Library” on page 247 to label an optical disk in an optical library.

Use the OAM LABEL command to label optical volumes on operator-accessible optical drives. OAM rejects the LABEL command if there is no operator-accessible optical drive online and operational.

Note: Because WORM optical volumes that are full or have very little free space are not useful as scratch volumes, the operator is notified, via a message, if the kilobytes free are less than the SCRENTRYTHRESHOLD parameter. The message contains the number of kilobytes that are free and the percentage of free space that this represents on the volume. This gives the operator the opportunity to fail the cartridge entry process, causing the cartridge to be ejected from the library.

In an OAMplex, if a shelf-resident volume is entered into a library that is not known to an OAM in the OAMplex, that OAM removes it from its configuration and issues a message that the volume is no longer known. After that time, any read for that volume on that OAM fails with a volume unknown error. If the volume is ejected and assigned back to a pseudo library that is known to OAM, the volume is added back to the configuration.

The following command syntax labels an optical disk on an operator-accessible optical drive:

```
►►──┬─MODIFY─┬──OAM,──┬─LABEL─┬─,─┬─3995REWR──┬─,─pseudo-library──────────────►◄
    └─F──────┘        └─L─────┘   ├─3995WORM──┤
                                  ├─3995-133──┤
                                  ├─3995-SW3──┤
                                  ├─3995-SW4──┤
                                  └─drive_name┘
```

The following describes the values of the options for the LABEL command:

3995REWR
refers to a request for a single-density, rewritable optical disk media to be labeled on an IBM 3995-131 operator-accessible optical disk drive.

3995WORM
refers to a request for a single-density, WORM optical disk media to be labeled on an IBM 3995-132 operator-accessible optical disk drive

3995-133
refers to a request for a single- or double-density, WORM or rewritable, optical disk media to be labeled on an IBM 3995-133 operator-accessible optical disk drive.

3995-SW3
refers to a request for a double- or quad-density capacity, WORM or rewritable optical disk media to be labeled on an IBM 3995-SW3 operator-accessible optical disk drive.

3995-SW4
refers to a request for a quad- or 8x-density capacity, WORM or rewritable optical disk media to be labeled on an IBM 3995-SW4 operator-accessible optical disk drive.

drive_name
refers to a IBM 3995 operator-accessible drive on which the label command is processed. The optical media being labeled must be compatible with the drive requested for reading and writing purposes. The name of the optical drive must be defined in the active SCDS configuration.

pseudo-library
refers to the name of the pseudo library whose drives are to be considered for the label request, and the pseudo library with which the newly labeled volume will be associated. If this keyword is not specified, the default will be the pseudo library associated with the operator-accessible drive that is performing the label function. If a pseudo library name specified on the LABEL command is invalid for the current ACDS, message CBR1305I is issued and the command fails.

Attention: Due to a hardware restriction, inserting a double-, quad-, or 8x-density, rewritable cartridge into a single-density, WORM (3995-132) or rewritable (3995-131) operator-accessible drive can result in the cartridge being demounted with no error message posted on the library service panel or to the host.

If the label request is directed to a specific drive, OAM verifies that the drive requested is currently managed and controlled by the OAM on which the command was entered. If this criteria is not satisfied, the request fails and message CBR1068I is issued.

If a pseudo-library name is specified on the LABEL command, a media type must be specified (3995REWR, 3995WORM, 3995-133, 3995-SW3, or 3995-SW4) because pseudo-libraries no longer require a device type affinity, so mixed devices and media types may be associated with a pseudo-library. If the drive name is specified, specification of a pseudo-library is ignored.

If only a media type is specified on the LABEL command, the command is processed on any operator-accessible drive that is capable of supporting the specified media type and that is connected to the OAM where the command was entered. Labels are processed on the system where the command is entered. The command must be issued on a system with an online and operational operator-accessible drive that can support the media type.

To write the label on two volumes of an optical disk on an operator-accessible drive:

1. Enter the following command:

```
F OAM,LABEL,{3995REWR|3995WORM|3995-133|3995-SW3|3995-SW4,drive_name},pseudo-lib
```

2. The system issues the following message:

```
CBR1000I OAM LABEL command execution scheduled.
```

3. When an operator-accessible optical drive becomes available, the system issues the following messages and waits for you to mount an optical disk cartridge:

```
CBR4400A Mount volume ?????? on drive drive-name.  Shelf location is ??????.
```

4. Put an unlabeled optical disk cartridge into the operator-accessible optical drive. The system issues the following message:

```
CBR3381I Volume mounted on drive drive-name contains an unrecognized format.
```

5. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive drive-name has unrecognized media
         format.  Reply 'F' to format or 'C' to cancel.
```

   Note: This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of ‘F’ causes the volume to be formatted and any data currently existing on the volume is destroyed. Further information about commands from the 3995 dynamic console can be found in the *3995 Operator’s Guide*.

6. To continue the label process, reply with an ‘F’. To cancel the label process, reply with an ‘C’. If a reply of ‘F’ is entered, the system issues the following message and waits for a reply:

```
CBR4405D Enter VOLSER for volume on drive drive-name.
```

7. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the

volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D Volser for unlabeled volume in drive drive-name is volser.
         Reply 'U' to use this volser, or 'R' to retry.
```

8. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4406D Enter owner information for volume volser on drive drive-name.
```

9. Provide 1 to 64 characters of owner identification. The system issues the following message and waits for a reply:

```
CBR4423D Enter shelf information for volume volser on drive drive-name.
```

10. Provide 1 to 32 characters of shelf information. The system issues the following message:

```
CBR4430A Remove and flip cartridge on drive drive-name.
```

11. Remove the optical disk cartridge, flip it over, and reinsert it into the operator-accessible optical drive. The system issues the following message:

```
CBR3381I Volume mounted on drive drive-name contains an unrecognized format.
```

12. The system issues the following message and waits for a reply:

```
CBR4438D Volume in drive drive-name has unrecognized media format. Reply 'F'
         to format or 'C' to cancel.
```

Note: This message is unique to a 3995 library. It is issued when a volume has been entered that is either unlabeled or of a format not known to the library. A reply of ‘F’ causes the volume to be formatted and any data currently existing on the volume is destroyed.

13. To continue the label process, reply with an ‘F’. To cancel the label process, reply with an ‘C’. If a reply of ‘F’ is entered, the system issues the following message and waits for a reply:

```
CBR4439D Enter volser for opposite side of volume volser in drive
         drive-name.
```

14. Provide a 1 to 6 character unique volume serial number. The system issues the following message to verify the volume serial number. To accept the

volume serial number, reply with a ‘U’. A reply of ‘R’ allows you to enter a different volume serial number.

```
CBR4424D Volser entered for unlabeled volume in drive drive-name is volser.
         Reply 'U' to use this volser, or 'R' to retry.
```

15. Once the system verifies the volume serial number to be used on the cartridge, the system issues the following message and waits for a reply:

```
CBR4432D Enter storage-group-name for volumes volser1 and volser2, or
         reply 'U' to assign to scratch.
```

16. If the volumes are to be assigned to scratch status, reply ‘U’ to this message; otherwise, reply with the name of the object storage group or object backup storage group to which the volumes are to be assigned. The system issues the following messages:

```
CBR4401I Volume volser-2 mounted on drive drive-name.

CBR2102I LABEL function complete for volumes volser-1 and volser-2.
```

The two volumes on the optical disk are added to a storage group or are assigned to scratch status and are available for use. In the message text, *volser-1* and *volser-2* are replaced by the volume serial numbers that you entered for the optical volumes.

Relabeling a 3995 Optical Disk Volume

The relabel command is used to allow the operator to rename a volume serial number on a previously defined 3995 optical disk volume. As a preventive measure to keep from losing all active primary or backup copies of objects on the 3995 disk volume that is being relabeled, the following conditions apply:

- The optical disk volume being relabeled cannot be an Object Backup volume.
- There can be no active primary copy of an object on the volume.
- There can be no write request currently scheduled for the volume.
- In an OAMplex environment, the requested volume must be currently managed and controlled by the OAM on which the command was entered.

To erase all copies of objects on the 3995 optical disk volume, submit the OAM utility job to reformat the volume. See “Reformatting a 3995 Optical Disk” on page 266 for more information on this utility.

The following command syntax relabels an optical disk volume:

```
►►──┬─MODIFY─┬──OAM,──┬─RELABEL─┬─,─old_volser─,─new_volser─,─┬────────────┬──────►◄
    └─F──────┘        └─RL──────┘                             └─drive_name─┘
```

The following describes the values of the options for the RELABEL command:

old_volser
: This a required parameter that specifies the current volume serial number of the OAM volume.

new_volser
: This is a required parameter that specifies the new volume serial number to be assigned to the optical volume.

drive_name
: This is an optional parameter that specifies a write compatible operator-accessible drive for processing this relabel request if the volume resides outside a 3995 optical disk library. If *drive_name* is not supplied or if the volume is library-resident, OAM selects an optical drive in the ACDS that is capable of processing this request.

To relabel a requested optical disk volume and direct the request to a specific operator-accessible drive, enter the following command:

```
F OAM,RELABEL,old_volser,new_volser,drive_name
```

This system issues the following messages:

```
CBR1000I OAM RELABEL command execution scheduled.

CBR4460I Volume old_volser on drive-name has been relabeled to new_volser.

CBR2822I RELABEL function completed for volume old_volser to new_volser.
```

When the command is accepted, if the requested volume is shelf-resident and not currently mounted on the selected drive, OAM asks the operator to mount the requested volume on the selected optical drive. If the requested volume is library-resident, OAM mounts the volume on a library drive. Following successful completion of this processing, OAM then performs the following functions:

- deletes the row in the DB2 Volume Table row for the old volume serial number
- inserts a new row in the DB2 Volume Table for the new volume serial number
- updates the row of the opposite volume in the DB2 Volume Table with the new volume serial number
- issues message CBR4460 to inform the operator that the relabeling of the 3995 optical disk volume has completed

Reformatting a 3995 Optical Disk

The OAM reformat utility (the TSO command OAMUTIL) is used to perform various tasks against a 3995 optical disk cartridge to reclaim usable space on the cartridge. This utility can be invoked either by running the CBRSAMUT SAMPLIB job, (see "CBRSAMUT" on page 434 for a sample job that can be used within your installation) or by issuing a TSO command to start the utility. This utility allows you to perform the following tasks:

- Reformat one or both sides of a 3995 optical disk cartridge.
- Reformat and rename the volume serial number of one or both sides of an optical disk cartridge.
- Return the volumes back to the SCRATCH storage group (only when there is a request for both sides of the optical disk to be reformatted) to be used for subsequent write requests.

The reformat utility can be run regardless of whether the volume is inside or outside a 3995 optical library, the volume belongs to an Object or an Object Backup storage group, or the media is rewritable or WORM. Reformatting rewritable media reclaims the used space; however, reformatting WORM media cannot reclaim the used space.

This utility can be run conditionally (using the NOFORCE parameter) or unconditionally (using the FORCE parameter).

When a *conditional* request is submitted, the volume is checked to see whether there are any backup copies or active primary copies of objects before performing the operation. If there are, the request fails for the reformat.

When an *unconditional* request is submitted, there is no verification of whether there are any non-expired objects still on the volume prior to the operation being performed. The volume is reinitialized if the OAM environment permits it at the time of the request. It is therefore recommended, that before submitting an unconditional reformat request, you should select all Object Directory entries for objects that reside on the requested volume (using SPUFI) to verify that the object can be deleted. Or, you can use the Move Volume utility to move all the objects off the requested volume before performing the reformat. Once an unconditional reformat request is executed successfully, all objects on the requested volume, regardless if the objects are owned by OAM, are all erased or discarded.

Attention: If you use the FORCE parameter or the unconditional form of the format command, be aware that once you request a reformat of the optical disk, there is no reversal. If OAM issues any error messages indicating DB2 or other problems, you are responsible for ensuring that all objects that resided on this volume are no longer referenced in the OAM object directory table.

For a reformat command to execute successfully, the following OAM environment must exist:

- No write, relabel, or reformat requests are currently scheduled for the requested volume.
- No eject request is currently scheduled (for the requested volume or its opposite side).
- The volume is not write protected and is not marked not writable.
- In an OAMplex environment, the requested volume is managed and controlled by the instance of OAM on which the reformat request was entered.

The following is the syntax for the OAMUTIL REFORMAT command:

```
                                     ┌─ONE──┐
►►──OAMUTIL──REFORMAT──old_volser────┴─BOTH─┴──┬──────────────────────────┬──────────►
                                               └─NEWVOL1(new_volser1)─────┘

                                                                ┌─NOSCRATCH─┐
►──┬──────────────────────────┬──┬──────────────────────────┬───┴─SCRATCH───┴────────►
   └─NEWVOL2(new_volser2)─────┘  └─DRIVENAME(drive_name)────┘

   ┌─NOFORCE─┐
►──┴─FORCE───┴───────────────────────────────────────────────────────────────────►◄
```

The following is a description of each of the keywords for this command:

old_volser
Indicates the volume serial number of the volume to be reformatted.

ONE | BOTH
An optional parameter. ONE indicates only one side of optical cartridge should be reformatted. This is the default.

BOTH indicates that both sides of the optical disk cartridge should be reformatted.

NEWVOL1(*new_volser1***)**
An optional parameter that indicates the new volume serial number for side one of the optical disk cartridge. If not specified, there is no change to the volume serial number of side one of the optical disk cartridge.

NEWVOL2(*new_volser2***)**
An optional parameter that indicates the new volume serial number for side two of the optical disk cartridge. If not specified, there is no change to the volume serial number of side two of the optical disk cartridge.

DRIVENAME(*drive_name***)**
An optional parameter that indicates a specified operator-accessible drive that is capable of processing this request. This parameter is only valid if the requested optical disk volume is shelf-resident. If this parameter is not specified, or if the requested optical disk volume is library-resident, OAM selects an optical drive in the SMS Active Configuration Data Set (ACDS) capable of processing this request.

NOSCRATCH | SCRATCH
An optional parameter that indicates which storage group the volume should be placed in after reformatting. NOSCRATCH indicates that the volume should remain in the same storage group to which it was assigned before the successful completion of the reformat execution. *This is the default.*

SCRATCH indicates that both volumes on an optical disk cartridge should be placed in the SCRATCH storage group on successful execution of the reformat request.

NOFORCE | FORCE
An optional parameter that indicates what type of reformat request is being run. NOFORCE indicates that the reformat request is conditional. The utility must verify the existence of any backup copies or active primary copies of objects on the volume before performing the reformat. If these objects exist on the volume, the reformat fails. *This is the default.*

FORCE indicates that the reformat request is unconditional. This parameter allows you to physically erase all copies of objects on a rewritable optical disk cartridge or discard all copies of objects on a WORM optical disk cartridge without first verifying if there are any backup copies or active primary copies of objects on the volume. The volume is reinitialized even if there are nonexpired objects on the volume.

To reformat a requested optical disk volume, enter the following TSO command:

```
OAMUTIL REFORMAT old_volser BOTH
```

This system issues the following messages:

```
CBR4401I Volume old_volser1 mounted on drive drive-name.

CBR4465I Volumes old_volser1 and old_volser2 are being reformatted on drive
         drive-name.

CBR4401I Volume old_volser2 mounted on drive drive-nam.

CBR4462I Volume old_volser1 has been reformatted to new_volser1.

CBR4462I Volume old_volser2 has been reformatted to new_volser2.
```

The following conditions exist upon successful completion of this command:

- All object directory entries for the requested volumes are deleted from the 3995 controller, the Object Directory Table, and the Deleted Objects Table.
- All space used by the erased objects are reclaimed for rewritable optical disk volumes.
- If SCRATCH is specified, both volumes on the optical disk cartridge are returned to the SCRATCH storage group.
- If NEWVOL1 is specified, a new row for this new volume serial number for the first side of the optical disk cartridge is added to the Volume Table. The row for the old volume serial number of the requested volume is deleted from the Volume Table. If NEWVOL1 is not specified, the row for the requested volume in the OAM Volume Table is updated.
- If NEWVOL2 is specified and it is a both side reformat, a new row for the new volume serial number for the second side of the optical disk cartridge is added to the Volume Table. The row in the Volume Table for the old volume serial number is deleted. If NEWVOL2 is not specified or the request is for a one side reformat, the row in the Volume Table for the opposite side of the requested volume is updated.

For information concerning messages generated from this command, use LookAt or see *MVS System Messages*. For a description of LookAt, see “Using LookAt to look up message explanations” on page xv.

Displaying Status

You can display the status of various items of the OAM system:

- OAM status
- OAM XCF status
- OSMC summary status

- OSMC task status
- Drive online/offline connectivity
- Drive detail status
- Library online/offline connectivity
- Library detail status
- Storage group and volume status
- SETOAM parameters
- SETOPT parameters

Note: In order to display information in the Tape Volume table concerning objects stored on tape volumes, you must use the SPUFI SELECT command. Using the DISPLAY command against these tape volumes only provides information from the Tape Configuration database, not the Tape Volume table.

Displaying OAM Status

The following command syntax displays OAM status:

```
►►──┬─DISPLAY─┬──SMS,OAM──┬───────────────────┬──────────────────────────►◄
    └─D───────┘           └─,L=──┬─a──────┬──┘
                                 ├─cc─────┤
                                 ├─cca────┤
                                 ├─name───┤
                                 └─name-a─┘
```

OAM
Displays OAM status.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console that is referred to by *name* and the screen that is referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display OAM status, enter the following command:

```
DISPLAY SMS,OAM
```

The OAM Display status shows which backup copy is being used by automatic access to backup processing for each of the possible reasons. If automatic access to backup is not active for a specific reason, then the status display shows that no backup copy is being used. The following information is displayed for an optical library. Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for an example of this display for a tape library.

```
CBR1100I  OAM status:
TOT   USE   TOT  USE  AVL  TOT  USE   AVL  TOT  USE  AVL  SCR   REQ
LIB   LIB   DRV  DRV  DRV  LDR  LDR   LDR  SDR  SDR  SDR  VOL    CT
aaa   bbb   ccc  ddd  eee  fff  ggg   hhh  iii  jjj  kkk  lll   mmm
exitname processing {Enabled|Disabled|Bypassed|Operator-Disabled}.
Access Backup status for xxx reasons, using yyy backup copy.
XCF GROUP NAME: group-name
XCF MEMBER NAME: member-name
```

The fields in the data line specify the number of each resource, as follows:

aaa Total number of optical libraries in the configuration.

bbb Number of usable optical libraries (online and operational).

ccc Total number of optical drives in the configuration.

ddd Number of usable optical drives.

eee Number of available optical drives (online, operational, and not currently in use).

fff Total number of library optical drives in the configuration.

ggg Number of usable library optical drives.

hhh Number of available library optical drives.

iii Total number of stand-alone or operator-accessible optical drives in the configuration.

jjj Number of usable stand-alone or operator-accessible optical drives.

kkk Number of available stand-alone or operator-accessible optical drives.

lll Number of scratch optical volumes in the optical configuration database.

mmm Total number of optical read requests waiting to be scheduled.

The following fields are displayed in the status message for the OAM access backup processing:

status The status of Access Backup processing for this reason. The following are valid *status* values:

- **ACTIVE**—Access Backup processing is active for this reason.
- **INACTIVE**—Access Backup processing is inactive for this reason.

xxx The reason for which Access Backup processing can be activated. The following are valid *xxx* values:

- **UNREADABLE VOLUMES**—The backup copy of an object is retrieved when the primary copy resides on a volume with READABLE=N.
- **OFFLINE LIBRARIES**—The backup copy of an object is retrieved when the primary copy resides on a volume that is in an offline library.
- **NOT OPERATIONAL LIBRARIES**—The backup copy of an object is retrieved when the primary copy resides on a volume that is in a library that is nonoperational.

yyy
The indicator of which backup copy, if any, is being used for automatic access to backup processing. The following are valid values for *yyy*:

- **1ST**—Access Backup processing accesses the first backup copy of the object when the primary copy is unavailable for the reason shown in *xxx*.
- **2ND**—Access Backup processing accesses the second backup copy of the object when the primary copy is invaluable for the reason shown in *xxx*.
- **NO**—Access Backup processing is inactive for the reason shown in *status*; therefore, no backup copy is being used.

group-name
The XCF group name for this OAMplex.

member-name
The XCF member name for this instance of OAM in an OAMplex.

The following is a sample of DISPLAY SMS,OAM status:

```
CBR1100I OAM status: 917
OPT. TOT  USE  TOT  USE  AVL  TOT  USE  AVL  TOT  USE  AVL  SCR  REQ
     LIB  LIB  DRV  DRV  DRV  LDR  LDR  LDR  SDR  SDR  SDR  VOL   CT
       6    1   27    5    5   22    4    4    5    1    1    6    0
TAPE TOT  ONL  TOT  TOT  TOT  TOT   TOT   ONL   AVL   TOTAL
     LIB  LIB   AL   VL  VCL   ML   DRV   DRV   DRV  SCRTCH
       2    1    0    1    0    0    16     2     2      21
There are also 0 VTS distributed libraries defined.
CBRUXCUA processing ENABLED.
CBRUXEJC processing ENABLED.
CBRUXENT processing ENABLED.
CBRUXVNL processing ENABLED.
CBRUXSAE processing ENABLED.
Access Backup processing INACTIVE for UNREADABLE VOLUMES, using no
backup copy.
Access Backup processing INACTIVE for OFFLINE LIBRARIES, using no
backup copy.
Access Backup processing INACTIVE for NOT OPERATIONAL LIBRARIES,
using no backup copy.
DB2 SSID:  DB2
XCF GROUP NAME: -N/A-
XCF MEMBER NAME: -N/A-
```

Displaying OAM XCF Status

This command displays system status for this instance of OAM in relation to the sysplex and XCF. The following command syntax displays OAM XCF status:

```
►►──┬─DISPLAY─┬──SMS,OAMXCF──────────────────────────────────────────►◄
    └─D───────┘
```

The following is a description of the keyword for this command:

OAMXCF
Displays OAM XCF status.

To display OAM XCF status, enter the following command:

```
DISPLAY SMS,OAMXCF
```

The following information is displayed:

```
CBR1250I OAM XCF STATUS:
XCF MEMBER NAME    USER              SYSTEM    OPT     OPT    TAPE
                   STATE             NAME      READ    WRITE  READ
xcf-member-name    aaaaaaaaaaaaaaaa  bbbbbbbb  cccc    ddddd  eeeee
-----------------------------------------------------------------------------

this-xcf-member   fffffffffffffff   gggggggg  hhhh    iiiii  jjjjj
XCF GROUP NAME:  xcf-group-name
OAM XCF timeout value for XCFOPTREADA is seconds.
OAM XCF timeout value for XCFOPTREADM is seconds.
OAM XCF timeout value for XCFOPTWRITEA is seconds.
OAM XCF timeout value for XCFOPTWRITEM is seconds.
OAM XCF timeout value for XCFTAPEREADA is seconds.
OAM XCF timeout value for XCFTAPEREADM is seconds.
```

For instances of OAM other than the OAM where the display command was issued, the fields displayed in each data line are as follows:

xcf-member-name
The member name associated with an instance of OAM in the OAMplex

aaaaaaaaaaaaaaaa
The user state of *xcf-member-name* on this data line. OAM defined user states are INITIALIZING, TERMINATING, RESTARTING, or ACTIVE.

bbbbbbbb
The system name associated with *xcf-member-name* on this data line.

cccc
The number of optical reads sent from the OAM where the command was entered to the OAM on the display line to be processed.

ddddd
The number of optical writes sent from the OAM where the command was entered to the OAM on the display line to be processed.

eeeee
The number of tape reads sent from the OAM where the command was entered to the OAM on the display line to be processed.

For instances of OAM on the system where the display command was issued, the following fields are displayed in the last data line of the multiline message:

this-xcf-member
The member name associated with this instance of OAM in the OAMplex where the display command was issued.

fffffffffffffff
User state of *this-xcf-member* where the command was issued. OAM defined user states are INITIALIZING, TERMINATING, RESTARTING, or ACTIVE.

gggggggg
System name associated with *xcf-member-name* on this data line.

hhhhh — The total number of optical reads sent from the OAM where the command was entered to other OAMs in the OAMplex.

iiiii — The total number of optical writes sent from the OAM where the command was entered to other OAMs in the OAMplex.

jjjjj — The total number of tape reads sent from the OAM where the command was entered to other OAMs in the OAMplex.

xcf-group-name — The XCF group associated with the OAMplex.

seconds — The OAM XCF timeout values for each XCFTIMEOUT subparameter (specified in the CBROAM*xx* member of PARMLIB when OAM was initialized or set by operator command) currently in effect for the OAM where the command was entered.

The following is a sample of DISPLAY SMS,OAMXCF status:

```
CBR1250I OAM XCF status:
XCF MEMBER NAME   USER          SYSTEM    OPT    OPT    TAPE
                  STATE          NAME     READ   WRITE  READ
OAMSYS2           ACTIVE        SYSTEM2     27      65     0
OAMSYS3           ACTIVE        SYSTEM3     36       0    22
------------------------------------------------------------
OAMSYS1           ACTIVE        SYSTEM1     63      65    22
XCF GROUP NAME:  OAMGRP1
OAM XCF timeout value for XCFOPTREADA is 20.
OAM XCF timeout value for XCFOPTREADM is 50.
OAM XCF timeout value for XCFOPTWRITEA is 150.
OAM XCF timeout value for XCFOPTWRITEM is 150.
OAM XCF timeout value for XCFTAPEREADA is 40.
OAM XCF timeout value for XCFTAPEREADM is 50.
```

If the instance of OAM is not part of an OAMplex, the following message will display:

```
CBR1069I Command rejected. OAM is not a member of an XCF group in a
         sysplex environment.
```

Displaying OSMC Summary Status

The following command syntax displays OSMC status:

```
►►──┬─DISPLAY─┬──SMS,OSMC──┬─────────────────┬──────────────────►◄
    └─D───────┘            └─,L=─┬─a──────┬─┘
                                 ├─cc─────┤
                                 ├─cca────┤
                                 ├─name───┤
                                 └─name-a─┘
```

OSMC
Displays OSMC status.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display OSMC summary status, enter the following command:

```
DISPLAY SMS,OSMC
```

The following information is displayed:

```
CBR9350I OSMC Summary Status:
TASK      TASK      TASK       START       OBJECTS       OBJECTS
NAME      TYPE      STAT       TIME        COMPLETED     ACTIVE
tskname   tsktype   tskstat    starttime   objcomplete   objactive
End of Display Summary
```

The fields displayed in each data line are as follows:

tskname Name of control task

tsktype Type of control task:
- **C** Cycle processing for a storage group
- **D** DASD space management
- **G** Operator requested processing of a single storage group
- **L** Library space management
- **M** Move Volume
- **R** Volume recovery

tskstat Task current status:
- **b̸** (Blank) OSMC is running
- **P** OSMC is stopping
- **T** OSMC is terminating

starttime Task start time (hh.mm.ss)

objcomplete Number of objects that have completed processing

objactive Number of objects currently being processed

The following is a sample of DISPLAY OSMC summary status:

```
CBR9350I OSMC Summary Status:
TASK       TASK      TASK       START          OBJECTS        OBJECTS
NAME       TYPE      STAT       TIME           COMPLETED      ACTIVE
GROUP02     G                   19.09.53        32732          3891
GROUP26     C                   19.09.55          480           266
GROUP27     C                   19.10.11         5890           329
VOL001      M                   19.10.14           59             8
End of Display Summary
```

Displaying OSMC Task Status

The following command syntax displays OSMC task status:

```
►►──┬─DISPLAY─┬──SMS,OSMC,TASK─(taskname)─┬──────────────┬──────────────►◄
    └─D───────┘                           └─,L=─┬─a──────┬─┘
                                                ├─cc─────┤
                                                ├─cca────┤
                                                ├─name───┤
                                                └─name-a─┘
```

To display the status of an OSMC task, enter the following command:

```
DISPLAY SMS,OSMC,TASK(taskname)
```

OSMC
Displays OSMC task status

TASK(*taskname***)**
Specifies the name of the task for which a status display is requested

The value of the task name depends on the type of OSMC process which the task represents. The following list shows the type of OSMC process, and the value used for its name.

Library space management
The library name

Volume recovery
The volume serial of one of the volumes on the disk being recovered

DASD space management
The name of the storage group being processed by DASD space management

OSMC processing of one storage group
The name of the storage group being processed

Storage management cycle processing of a storage group
The name of the storage group being processed

Move volume
The volume serial of the source volume from which objects are being moved

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

NOT Programming Interface information

Detail status information is provided for the OAM storage management component task specified in the DISPLAY command. The number of internal work items queued on the work and wait queues and the number of internal work items completed for each of the OAM storage management component (OSMC) services is displayed. The number of internal work items does not exactly equate to the number of objects processed; there may be multiple internal work items per object or there may be internal work items not associated with any object. This information is better used for problem determination and monitoring the progress of the storage management component than for tracking the actual number of objects processed.

End of NOT Programming Interface information

The fields displayed in each data line represent the services that OSMC performs during its processing. The following information is displayed:

```
CBR9370I OSMC Detail for taskname:
          READ   READ   READ  WRITE  WRITE
          DASD    OPT   TAPE   DASD    OPT
WORK Q: aaaaaa bbbbbb cccccc dddddd eeeeee
WAIT Q:        jjjjjj kkkkkk        llllll
DONE:   qqqqqq rrrrrr ssssss tttttt uuuuuu
          WRITE  WRITE    WRITE      DIR
          TAPE   BACKUP1  BACKUP2  UPDTS
WORK Q: ffffff   gggggg   hhhhhh   iiiiii
WAIT Q: mmmmmm   nnnnnn   oooooo   pppppp
DONE:   vvvvvv   wwwwww   xxxxxx   yyyyyy
End of Display Detail
```

In the message text, *taskname* is the name associated with the OAM storage management component task and is the same as the task name specified on the DISPLAY SMS,OSMC command. In the case of the OAM storage management cycle, *taskname* is the name of an Object storage group being processed by OSMC. In the case of the OAM MOVEVOL utility, *taskname* is the volume serial number of the volume being operated on by the utility. In the case of the OAM Volume Recovery utility, *taskname* is the volume serial number of the optical or tape cartridge being recovered by the utility. For an example of output for message CBR9370I, see page 278.

The following are descriptions of the column headings for CBR9370I:

READ DASD The READ DASD column contains the number of internal work items that are queued on the work queue and the number of internal work items that are completed by the read DASD service.

READ OPT The READ OPT column contains the number of internal work items that are queued on the work and wait queues and the number of internal work items that are completed by the read optical service.

READ TAPE The READ TAPE column contains the number of internal work items that are queued on the work and wait queues and the number of internal work items that are completed by the read tape service.

WRITE DASD The WRITE DASD column contains the number of internal work items that are queued on the work queue and the number of internal work items that are completed by the write DASD service.

WRITE OPT The WRITE OPT column contains the number of internal work items that are queued on the work and wait queues and the number of internal work items that are completed by the write optical service.

WRITE TAPE The WRITE TAPE column contains the number of internal work items that are queued on the work and wait queues and the number of internal work items that are completed by the write tape service.

WRITE BACKUP1
The WRITE BACKUP1 column contains the number of internal work

items that are queued on the work and wait queues by the write first backup service and the number of internal work items that are completed by the write first backup service.

WRITE BACKUP2
The WRITE BACKUP2 column contains the number of internal work items that are queued on the work and wait queues by the write second backup service and the number of internal work items that are completed by the write second backup service.

DIR UPDTS
The DIR UPDTS column contains the number of internal work items that are queued on the work and wait queues and the number of internal work items that are completed by the directory update service.

The fields that are displayed in each row represent the number of internal work items (*n*) that are currently at that stage of processing for each service:

WORK Q
Work items queued for processing by this service

WAIT Q
Work items for which processing has started but not completed

DONE
Work items that have completed using this service

The values in the display for CBR9370I are defined as follows:

aaaaaa
The number of internal work items that are queued on the work queue by the read DASD service.

bbbbbb
The number of internal work items that are queued on the work queue by the read optical service.

cccccc
The number of internal work items that are queued on the work queue by the read tape service.

dddddd
The number of internal work items that are queued on the work queue by the write DASD service.

eeeeee
The number of internal work items that are queued on the work queue by the write optical service.

ffffff
The number of internal work items that are queued on the work queue by the write tape service.

gggggg
The number of internal work items that are queued on the work queue by the write first backup service.

hhhhhh
The number of internal work items that are queued on the work queue by the write second backup service.

iiiiii
The number of internal work items that are queued on the work queue by the directory update service.

jjjjjj
The number of internal work items that are queued on the wait queue by the directory update service.

kkkkkk
The number of internal work items that are queued on the wait queue by the read tape service.

llllll
The number of internal work items that are queued on the wait queue by the write optical service.

mmmmmm
The number of internal work items that are queued on the wait queue by the write tape service.

nnnnnn The number of internal work items that are queued on the wait queue by the write first backup service.

oooooo The number of internal work items that are queued on the wait queue by the write second backup service.

pppppp The number of internal work items that are queued on the wait queue by the directory update service.

qqqqqq The number of internal work items that are completed by the read DASD service.

rrrrrr The number of internal work items that are completed by the read optical service.

ssssss The number of internal work items that are completed by the read tape service.

tttttt The number of internal work items that are completed by the write DASD service.

uuuuuu The number of internal work items that are completed by the write optical service.

vvvvvv The number of internal work items that are completed by the write tape service.

wwwwww The number of internal work items that are completed by the write first backup service.

xxxxxx The number of internal work items that are completed by the write second backup service.

yyyyyy The number of internal work items that are completed by the directory update service.

The following is a sample of DISPLAY SMS,OSMC,TASK(WG360A):

```
CBR9370I OSMC Detail for WG360A:
          READ   READ   READ  WRITE  WRITE
          DASD    OPT   TAPE   DASD    OPT
WORK Q:      0      1      0      0      0
WAIT Q:             3      0             1
DONE:        0      1      1      0      1
          WRITE  WRITE   WRITE   DIR
          TAPE   BACKUP1 BACKUP2 UPDTS
WORK Q:      0      0      0      0
WAIT Q:      0      0      0      0
DONE:        0      0      0      1
End of Display Detail
```

Displaying Drive Online/Offline Connectivity

The following command syntax displays OAM drive status:

```
►►──┬─DISPLAY─┬──SMS,─┬─DRIVE─┬─(─┬─name─┬─)─,STATUS─┬──────────────┬──────────►◄
    └─D───────┘       └─DRI───┘   └─ALL──┘            └─,L=─┬─a──────┤
                                                             ├─cc─────┤
                                                             ├─cca────┤
                                                             ├─name───┤
                                                             └─name-a─┘
```

DRIVE(*name* | ALL)
Displays the system connectivity and online/offline status for optical drives. If a drive name is specified, there is one data line describing the specified optical drive. If ALL is specified, all the optical drives in the SMS configuration are displayed. To specify a drive named ALL, surround it with parentheses; for example, DISPLAY SMS,DRIVE((ALL)).

STATUS
Displays the system connectivity and online/offline status.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display SMS,DRIVE,STATUS for an individual drive, enter the following command:

```
DISPLAY SMS,DRIVE(drvname),STATUS
```

The following information is displayed:

```
IGD002I hh.mm.ss DISPLAY SMS
                                                    1 1 1 1 1 1 1
DRIVE      LIBRARY      SYSTEM=   1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
name       aaaaaaaa               s s s s s s s s s s s s s s s s


                                  1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE      LIBRARY      SYSTEM=   7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
name       aaaaaaaa               s s s s s s s s s s s s s s s s
```

The fields displayed in each data line are as follows:

name — Name of the optical drive for which system connectivity and the online/offline status is displayed

aaaaaaaa — Name of the optical library to which the displayed drive belongs

1–32 — Numbers that appear after SYSTEM= indicating the system IDs

s — Indications of drive status:
- **.** Not defined
- **+** Online
- **−** Offline

Note: An Online status of “+” does not necessarily mean that either the drive or the library is fully functional. To determine if the drive is both online and operational, you must issue the DISPLAY SMS,DRIVE(*drvname*),DETAIL command. See “Displaying Drive Detail Status” on page 281 for more information on this SMS command. To determine if the library is both online and operational, you must issue the DISPLAY SMS,LIBRARY(*name*),DETAIL command. See “Displaying Library Detail Status” on page 290 for more information on this SMS command.

The following is a sample of DISPLAY SMS,DRIVE(*drvname*),STATUS:

```
D SMS,DRIVE(P13D1),STATUS
IGD002  15:08:16 DISPLAY SMS 397

                                                   1 1 1 1 1 1 1
DRIVE       LIBRARY      SYSTEM=1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
P13D1       PEA13               + - . . . . . . . . . . . . . .

                                1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE       LIBRARY      SYSTEM=7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1       PEA13               . . . . . . . . . . . . . . . .
**********************************LEGEND*************************
. THE DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE DRIVE IS ONLINE
- THE DRIVE IS OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,DRIVE(ALL),STATUS:

```
D SMS,DRIVE(ALL),STATUS
IGD002I 15:08:33 DISPLAY SMS 400
                                             1 1 1 1 1 1 1
DRIVE    LIBRARY SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
L1D0     LIB1            - + . . . . . . . . . . . . . .
L2D0     LIB2            - + . . . . . . . . . . . . . .
L3D0     LIB3            + - . . . . . . . . . . . . . .
L4D0     LIB4            + - . . . . . . . . . . . . . .
P13D1    PEA13           + - . . . . . . . . . . . . . .
P13D5    PCTREUSE        + - . . . . . . . . . . . . . .
P15D1    PEA15           + - . . . . . . . . . . . . . .
P15D5    PCTREUSE        + - . . . . . . . . . . . . . .
P17D1    PEA17           + - . . . . . . . . . . . . . .
P19D1    PEA19           + - . . . . . . . . . . . . . .
P21D1    PMA21           - + . . . . . . . . . . . . . .
P21D5    P3995133        - + . . . . . . . . . . . . . .
P7D1     PWA7            + - . . . . . . . - . . . . . .
P8D1     PWB8            + - . . . . . . . - . . . . . .
P9D1     PWA9            + - . . . . . . . . . . . . . .
SOU0     STDALONE        . . . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE    LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
L1D0     LIB1            . . . . . . . . . . . . . . . .
L2D0     LIB2            . . . . . . . . . . . . . . . .
L3D0     LIB3            . . . . . . . . . . . . . . . .
L4D0     LIB4            . . . . . . . . . . . . . . . .
P13D1    PEA13           . . . . . . . . . . . . . . . .
P13D5    PCTREUSE        . . . . . . . . . . . . . . . .
P15D1    PEA15           . . . . . . . . . . . . . . . .
P15D5    PCTREUSE        . . . . . . . . . . . . . . . .
P17D1    PEA17           . . . . . . . . . . . . . . . .
P19D1    PEA19           . . . . . . . . . . . . . . . .
P21D1    PMA21           . . . . . . . . . . . . . . . .
P21D5    P3995133        . . . . . . . . . . . . . . . .
P7D1     PWA7            . . . . . . . . . . . . . . . .
P8D1     PWB8            . . . . . . . . . . . . . . . .
P9D1     PWA9            . . . . . . . . . . . . . . . .
SOU0     STDALONE        . . . . . . . . . . . . . . . .
***************************** LEGEND ****************************
. THE DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE DRIVE IS ONLINE
- THE DRIVE IS OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

Displaying Drive Detail Status

The following command syntax displays OAM drive detail status:

```
►►──┬─DISPLAY─┬──SMS,──┬─DRIVE─┬──(──┬─drvname─┬──)─,DETAIL──┬──────────────┬──────►◄
    └─D───────┘        └─DRI───┘     ├─libname─┤             └─,L=──┬─a──────┤
                                     └─ALL─────┘                    ├─cc─────┤
                                                                    ├─cca────┤
                                                                    ├─name───┤
                                                                    └─name-a─┘
```

DRIVE(*drvname* | *libname* | ALL)
Displays the details of the current status for optical drives. If a drive name is specified, there is one data line describing the specified optical drive. If a library name is specified, there is a data line describing each optical drive associated with the library specified. If ALL is specified, all the optical drives in the SMS configuration are displayed. To specify a drive named ALL, surround it with parentheses, for example, DISPLAY SMS,DRIVE((ALL)).

DETAIL
Displays detail status for optical drives.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display detail status for an individual drive, enter the following command:

```
DISPLAY SMS,DRIVE(drvname),DETAIL
```

The following information is displayed:

```
CBR1120I OAM drive status:
DRIVE    DEVICE TY  LIBRARY  ON OP AV WP   DEV  SC  DRV  MOUNT   PEND
NAME     TYPE       NAME                   NUM  SI       VOLUME   VOLUME
drvname  devtype a  libname   b  c  d  e  ffff   g  hhh  mntvol  pndvol
--------------------------------------------------------
XCF MEMBER NAME: membername
--------------------------------------------------------
```

The fields displayed in each data line are as follows:

drvname
Name of the optical drive

devtype
Device type of optical drive:

3995-131	3995 rewritable optical disk drive
3995-132	3995 write-once optical disk drive
3995-133	3995 multifunction optical disk drive
3995-SW3	3995 multifunction optical disk drive
3995-SW4	3995 multifunction optical disk drive
9247	9246 library drive

a
Optical drive type:
- **L** Library
- **S** Stand-alone or operator-accessible

libname
Name of the library to which the optical drive is attached. For a stand-alone or an operator-accessible optical drive, this field displays the name you specified for your pseudo library, or the following as the defaults:
- **STDALONE** (for 9246 drives)
- **PCTWORM** (for 3995-132 write-once drives)
- **PCTREUSE** (for 3995-131 rewritable drives)
- **P3995133** (for 3995-133 multifunction drives)
- **P3995SW3** (for 3995-SW3 drives)
- **P3995SW4** (for 3995-SW4 drives)

b Optical drive online status:
- **Y** Online
- **N** Offline
- **P** Pending offline

c Optical drive operational status:
- **Y** Operational
- **N** Not operational

d Optical drive availability status:
- **Y** Available. The optical drive is online, operational, not pending offline, and not busy.
- **N** Not available.

e Write Protection status:
- **Y** Write protection is on. Writing to this drive is not allowed.
- **N** Write protection is off. Writing to this drive is allowed.

  The write protection status reflects the switch setting as of the last volume mount, vary online, or drive error processing.

ffff MVS/ESA device number of the CTC that is used to communicate with the optical drive.

g SCSI bus address of the optical drive on the SCSI interface. Not used for 3995 and will be blank.

hhh Drive number of the optical disk drive.

mntvol Volume serial number of the volume that is currently mounted on the optical drive. If there is no mounted volume, this field is blank.

pndvol Volume serial number of the volume for which a mount is pending on the 9247 drive. If there is no pending mount, this field is blank.

membername The XCF member name associated with the instance of OAM where this drive is currently online.

The following is a sample of DISPLAY SMS,DRIVE(LID1),DETAIL:

```
CBR1120I OAM drive status:
DRIVE     DEVICE   TY  LIBRARY  ON  OP  AV  WP  DEV  SC  DRV  MOUNT   PEND
NAME      TYPE         NAME                     NUM  SI  NUM  VOLUME  VOLUME
L1D1      3995-133  L  LIB1     Y   Y   N   N   0922        1  OP001

----------------------------------------------------
XCF MEMBER NAME: OAMXCFMEMBER1
----------------------------------------------------
```

To display detail status for drives associated with a specific optical library, enter the following command:

```
DISPLAY SMS,DRIVE(libname),DETAIL
```

The following is a sample of DISPLAY SMS,DRIVE(LIB1),DETAIL:

```
CBR1120I OAM drive status:
DRIVE  DEVICE   TY  LIBRARY  ON  OP  AV  WP  DEV  SC  DRV  MOUNT   PEND
NAME   TYPE         NAME                     NUM  SI  NUM  VOLUME  VOLUME
L1D1   3995-133  L  LIB1      Y   Y   N   N 0922        1  OP001
L1D2   3995-133  L  LIB1      Y   Y   N   N 0923        2  OP002
L1D3   3995-133  L  LIB1      Y   Y   Y   N 0924        3  OP003
L1D4   3995-133  L  LIB1      Y   Y   Y   N 0925        4  OP004
```

To display detail status for all the optical disk drives active in the SMS configuration, enter the following command:

```
DISPLAY SMS,DRIVE(ALL),DETAIL
```

The following is a sample of DISPLAY SMS,DRIVE(ALL),DETAIL:

```
CBR1120I OAM drive status:
DRIVE  DEVICE   TY  LIBRARY  ON  OP  AV  WP  DEV  SC  DRV  MOUNT   PEND
NAME   TYPE         NAME                     NUM  SI       VOLUME  VOLUME
L1D1   3995-133  L  LIB1      N   N   N   N 0992        1
L1D2   3995-133  L  LIB1      Y   Y   N   N 0993        2  OP002
L1D3   3995-133  L  LIB1      Y   Y   Y   N 0994        3  OP003
L1D4   3995-133  L  LIB1      Y   Y   Y   N 0995        4  OP004
L1D5   3995-133  S  PSEUDO1   N   N   N   N 0996        5
L2AD1  3995-SW3  L  LIB2      Y   Y   Y   N 19F6        1  OP012
L2AD2  3995-SW3  L  LIB2      N   Y   N   N 19F7        2
L2AD3  3995-SW3  L  LIB2      N   Y   N   N 19F8        3
L2AD4  3995-SW3  L  LIB2      N   Y   N   N 19F9        4
L2AD5  3995-SW3  L  LIB2      N   Y   N   N 19FA        5
L2AD6  3995-SW3  L  LIB2      N   Y   N   N 19FB        6
L3D1   3995-SW3  S  P3995SW3  N   Y   N   N 19F2        1
L3D1   3995-113  L  LIB2      Y   Y   N   N 092A        1  OP006
L3D2   3995-113  L  LIB2      Y   Y   N   N 092B        2  OP007
L3D3   3995-113  L  LIB2      Y   Y   Y   N 092C        3  OP008
L3D4   3995-113  L  LIB2      Y   Y   Y   N 092D        4  OP009
```

Displaying Library Online/Offline Connectivity

The following command syntax display OAM library status:

```
►►──┬─DISPLAY─┬──────────────────────────────────────────────────────────►
    └─D───────┘

►─SMS,──┬─LIBRARY─┬──(──┬─name─┬──)──┬──────────────────────────┬──┬──────────────┬──────►◄
        └─LIB─────┘     └─ALL──┘     ├─,STATUS──┬───────────┬───┤  └─,L=──┬─a──────┤
                                     │          └─,LISTDRI──┘   │         ├─cc─────┤
                                     └─,LISTDRI─────────────────┘         ├─cca────┤
                                                                          ├─name───┤
                                                                          └─name-a─┘
```

LIBRARY(*name* | ALL)
Displays the system connectivity and online/offline status for real libraries. If a library name is specified, there is one data line describing the specified library. If ALL is specified, there is one data line for each optical library in the configuration. To specify a library named ALL, surround it with parentheses; for example, DISPLAY SMS,LIBRARY((ALL)).

This display command applies only to real optical libraries. If the specified library is a pseudo optical library, an error message is displayed.

STATUS
Displays the system connectivity and online/offline status.

LISTDRI
Displays the online/offline status of all drives associated with this library.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display status for an individual library, enter the following command:

```
DISPLAY SMS,LIBRARY(libname),STATUS
```

The following information is displayed:

```
IGD002I 11.19.56    DISPLAY SMS
                                               1 1 1 1 1 1 1
LIBRARY  CLASS      SYSTEM=    1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
name     type                  s s s s s s s s s s s s s s s s

                               1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY  CLASS      SYSTEM=    7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
name     type                  s s s s s s s s s s s s s s s s
```

The fields displayed in each data line are as follows:

name Name of the library for which system connectivity and online/offline status is displayed

type Library type (optical or tape)

1–32 Numbers that appear after SYSTEM= indicating system IDs

s Indications of drive status:

- **.** Not defined
- **+** Online
- **−** Offline

Note: An Online status of “+” does not necessarily mean that either the drive or the library is fully functional. To determine if the drive is both online and operational, you must issue the DISPLAY SMS,DRIVE(*drvname*),DETAIL command. See “Displaying Drive Detail Status” on page 281 for more information on this SMS command. To determine if the library is both online and operational, you must issue the DISPLAY SMS,LIBRARY(*name*),DETAIL command. See “Displaying Library Detail Status” on page 290 for more information on this SMS command.

The following is a sample of DISPLAY SMS,LIBRARY(*libname*),STATUS:

```
D SMS,LIBRARY(PEA13),STATUS
IGD002I 15:09:05 DISPLAY SMS 403

                                            1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
PEA13     OPTICAL        + - . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
PEA13     OPTICAL        . . . . . . . . . . . . . . . .
***************************** LEGEND ****************************
. THE LIBRARY IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY IS ONLINE
- THE LIBRARY IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),STATUS:

```
D SMS,LIBRARY(ALL),STATUS
IGD002I 15:09:21 DISPLAY SMS 409
                                             1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
ATLF4017  TAPE           + - . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           - + . . . . . . . . . . . . . .
MTLA0002  TAPE           + - . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        + - . . . . . . . . . . . . . .
PEA15     OPTICAL        + - . . . . . . . . . . . . . .
PEA17     OPTICAL        + - . . . . . . . . . . . . . .
PEA19     OPTICAL        + - . . . . . . . . . . . . . .
PMA21     OPTICAL        + - . . . . . . . . . . . . . .
PWA7      OPTICAL        . . . . . . . . . + . . . . . .
PWA9      OPTICAL        + - . . . . . . . . . . . . . .
PWB8      OPTICAL        . . . . . . . . . - . . . . . .
P3995133  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
ATLF4017  TAPE           . . . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           . . . . . . . . . . . . . . . .
MTLA0002  TAPE           . . . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        . . . . . . . . . . . . . . . .
PEA15     OPTICAL        . . . . . . . . . . . . . . . .
PEA17     OPTICAL        . . . . . . . . . . . . . . . .
PEA19     OPTICAL        . . . . . . . . . . . . . . . .
PMA21     OPTICAL        . . . . . . . . . . . . . . . .
PWA7      OPTICAL        . . . . . . . . . . . . . . . .
PWA9      OPTICAL        . . . . . . . . . . . . . . . .
PWB8      OPTICAL        . . . . . . . . . . . . . . . .
P3995133  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
***************************** LEGEND ****************************
. THE LIBRARY IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY IS ONLINE
- THE LIBRARY IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIB(PEA13),LISTDRI:

```
D SMS,LIB(PEA13),LISTDRI
IGD002I 15:09:47 DISPLAY SMS 412
                                            1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
PEA13     OPTICAL        + - . . . . . . . . . . . . . .

                                            1 1 1 1 1 1 1
DRIVE    LIBRARY SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
P13D1    PEA13           + - . . . . . . . . . . . . . .
P13D2    PEA13           + - . . . . . . . . . . . . . .
P13D3    PEA13           + - . . . . . . . . . . . . . .
P13D4    PEA13           + - . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
LIBRARY   CLASS  SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
PEA13     OPTICAL        . . . . . . . . . . . . . . . .

                         1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE    LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1    PEA13           . . . . . . . . . . . . . . . .
P13D2    PEA13           . . . . . . . . . . . . . . . .
P13D3    PEA13           . . . . . . . . . . . . . . . .
P13D4    PEA13           . . . . . . . . . . . . . . . .
***************************** LEGEND ****************************
. THE LIBRARY OR DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY OR DRIVE IS ONLINE
- THE LIBRARY OR DRIVE IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),LISTDRI:

```
D SMS,LIBRARY(ALL),LISTDRI
IGD002I 15:09:21 DISPLAY SMS 409
                                            1 1 1 1 1 1 1
LIBRARY   CLASS  SYSTEM= 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
ATLF4017  TAPE           + - . . . . . . . . . . . . . .
LIB1      OPTICAL        . . . . . . . . . . . . . . . .
LIB2      OPTICAL        . . . . . . . . . . . . . . . .
LIB3      OPTICAL        . . . . . . . . . . . . . . . .
LIB4      OPTICAL        . . . . . . . . . . . . . . . .
MTLA0001  TAPE           - . . . . . . . . . . . . . . .
MTLA0002  TAPE           + - . . . . . . . . . . . . . .
PCTREUSE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PCTWORM   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
PEA13     OPTICAL        + - . . . . . . . . . . . . . .
PEA15     OPTICAL        + - . . . . . . . . . . . . . .
PWA9      OPTICAL        + - . . . . . . . . . . . . . .
P156A     OPTICAL        + - . . . . . . . . . . . . . .
3995133   OPTICAL                 LIBRARY IS NOT A REAL LIBRARY
STDALONE  OPTICAL                 LIBRARY IS NOT A REAL LIBRARY

                          1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
DRIVE     LIBRARY SYSTEM= 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
P13D1     PEA13           + - . . . . . . . . . . . . . .
P13D2     PEA13           + - . . . . . . . . . . . . . .
P13D3     PEA13           + - . . . . . . . . . . . . . .
P13D4     PEA13           + - . . . . . . . . . . . . . .
P15D1     PEA15           + - . . . . . . . . . . . . . .
P15D2     PEA15           + - . . . . . . . . . . . . . .
P15D3     PEA15           + - . . . . . . . . . . . . . .
P15D4     PEA15           + - . . . . . . . . . . . . . .
P9D1      PWA9            + - . . . . . . . . . . . . . .
P9D2      PWA9            + - . . . . . . . . . . . . . .
P9D3      PWA9            + - . . . . . . . . . . . . . .
P9D4      PWA9            P . . . . . . . . . . . . . . .
P156AD1   P156A           + - . . . . . . . . . . . . . .
P156AD2   P156A           + . . . . . . . . . . . . . . .
P156AD3   P156A           - + . . . . . . . . . . . . . .
P156AD4   P156A           - + . . . . . . . . . . . . . .
P156AD5   P156A           - + . . . . . . . . . . . . . .
P156AD6   P156A           - + . . . . . . . . . . . . . .
LISTDRI IS IGNORED FOR PSEUDO AND TAPE LIBRARIES
***************************** LEGEND *****************************
. THE LIBRARY OR DRIVE IS NOT DEFINED TO THE SYSTEM
+ THE LIBRARY OR DRIVE IS ONLINE
- THE LIBRARY OR DRIVE IS OFFLINE
P THE LIBRARY IS PENDING OFFLINE
SYSTEM 1  = SYSTEM1    SYSTEM 2  = SYSTEM2    SYSTEM 3  = SYSTEM3
SYSTEM 4  = SYSTEM4    SYSTEM 5  = SYSTEM5    SYSTEM 6  = SYSTEM6
SYSTEM 7  = SYSTEM7    SYSTEM 8  = SYSTEM8    SYSTEM 9  = SYSTEM9
SYSTEM 10 = SYSTEM10   SYSTEM 11 = SYSTEM11   SYSTEM 12 = SYSTEM12
SYSTEM 13 = SYSTEM13   SYSTEM 14 = SYSTEM14   SYSTEM 15 = SYSTEM15
SYSTEM 16 = SYSTEM16   SYSTEM 17 = SYSTEM17   SYSTEM 18 = SYSTEM18
SYSTEM 19 = SYSTEM19   SYSTEM 20 = SYSTEM20   SYSTEM 21 = SYSTEM21
SYSTEM 22 = SYSTEM22   SYSTEM 23 = SYSTEM23   SYSTEM 24 = SYSTEM24
SYSTEM 25 = SYSTEM25   SYSTEM 26 = SYSTEM26   SYSTEM 27 = SYSTEM27
SYSTEM 28 = SYSTEM28   SYSTEM 29 = SYSTEM29   SYSTEM 30 = SYSTEM30
SYSTEM 31 = SYSTEM31   SYSTEM 32 = SYSTEM32
```

Displaying Library Detail Status

The syntax of the DISPLAY SMS command for detail library status is:

```
►►──┬─DISPLAY─┬──SMS,─┬─LIBRARY─┬─(─┬─name─┬─)─,DETAIL─┬──────────────┬──────►◄
    └─D───────┘       └─LIB─────┘   └─ALL──┘           └─,L=─┬─a──────┬─┘
                                                             ├─cc─────┤
                                                             ├─cca────┤
                                                             ├─name───┤
                                                             └─name-a─┘
```

LIBRARY(*name* | ALL)
: Specifies the name of the optical library to be displayed. If a library name is specified, there is one data line describing the specified library. If ALL is specified, there is one data line for each optical library in the configuration. To specify a library named ALL, surround it with parentheses; for example, DISPLAY SMS,LIBRARY((ALL)).

DETAIL
: Displays detail status for optical libraries.

L={a | cc | cca | name | name-a}
: Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display detail status for an individual library, enter the following command:

```
DISPLAY SMS,LIBRARY(name),DETAIL
```

The following information is displayed:

```
CBR1110I OAM library status:
OPTICAL  DEVICE  ATT  USE  AVL  TOT  EMP  SCR  PT  DEV  ON  OP  IO   LIB    READ
LIBRARY  TYPE    DRV  DRV  DRV  SLT  SLT  VOL      NUM          ST   CMD   COUNT
olibname devtype aaa  bbb  ccc  ddd  eee  fff   g  hhhh  i   j   k   lbcmd rdcnt
DEFAULT PSEUDO LIB: def-plib-name
DEFAULT MEDIA TYPE: def-mediatype
XCF MEMBER NAME: membername
```

The fields displayed in each data line are as follows:

olibname
: Name of the optical library

devtype
: Device type of the optical library:

3995-111	3995 rewritable library, extension to a 3995-131
3995-112	3995 WORM library, extension to a 3995-132
3995-113	3995 multi-function library, extension to a 3995-133
3995-132	3995 WORM library and controller
3995-133	3995 multi-function library and controller
3995-C3A	3995 multi-function library controller
3995-C12	3995-multi-function library, extension to a 3995-C32
3995-C16	3995 multi-function library, extension to a 3995-C36
3995-C18	3995 multi-function library, extension to a 3995-C38
3995-C32	3995 multi-function library, attaches to a 3995-C3A

3995-C34	3995 multi-function library, attaches to a 3995-C3A
3995-C36	3995 multi-function library, attaches to a 3995-C3A
3995-C38	3995 multi-function library, attaches to a 3995-C3A
9246	IBM 9246 optical disk library

aaa
Number of optical drives attached to the optical library

bbb
Number of usable optical drives (online, operational, and not pending offline)

ccc
Number of available optical drives (online, operational, and not currently in use)

ddd
Total number of storage slots in the optical library

eee
Number of empty storage slots in the optical library

fff
Number of scratch volumes in the optical library

g
Active path to the optical library:

- **P** Primary
- **A** Alternate
- **BLANK** Pseudo library or 3995 library

hhhh
MVS/ESA device number of the active CTC, or blank for pseudo libraries

i
Optical library online status:

- **Y** Online
- **N** Offline
- **P** Pending offline

j
Optical library operational status:

- **Y** Operational
- **N** Not operational

k
Optical library input/output station operational status, as follows:

- **Y** Operational
- **N** Not operational
- ***** An error occurred while trying to get status
- **BLANK**
  Library not attached or library has no I/O station

lbcmd
For the 3995: (except for 3995-C3A) REMAP indicates that a REMAP of the library is in progress, RMPND indicates that a full library audit is being processed, and AUDIT indicates that a full library audit of the library is in progress. If not REMAP, RMPND, or AUDIT, this field contains the library command most recently sent to the optical library.

rdcnt
The number of read requests waiting or in progress for optical volumes that are resident in this optical library.

def-plib-name
The name of the pseudo library that will be assigned to any volume that is ejected from this library if that volume does not already have a pseudo library associated with it.

def-mediatype
The media types that can be entered into the optical library and which media types can be written to if they already reside in the library. This value is specified on the 3995 Library Define panel in ISMF.

membername
The XCF member name associated with the instance of OAM where this library is currently online.

The following is a sample of DISPLAY SMS,LIBRARY(LIB1),DETAIL:

```
CBR1110I OAM library status:
OPTICAL  DEVICE    ATT  USE  AVL  TOT  EMP  SCR  PT  DEV   ON  OP  IO  LIB   READ
LIBRARY  TYPE      DRV  DRV  DRV  SLT  SLT  VOL       NUM          ST  CMD   COUNT
LIB1     3995-C36   6    1    1  156  153    4     19F4    Y   Y   Y  LM        0
-------------------------------------------------------
DEFAULT PSEUDO LIB: PLIB4
DEFAULT MEDIA TYPE: 3995REWR
XCF MEMBER NAME: OAMSYS1
-------------------------------------------------------
```

The following is a sample of DISPLAY SMS,LIBRARY(ALL),DETAIL:

```
CBR1110I OAM library status:
OPTICAL  DEVICE    ATT  USE  AVL  TOT  EMP  SCR  PT  DEV   ON  OP  IO  LIB   READ
LIBRARY  TYPE      DRV  DRV  DRV  SLT  SLT  VOL       NUM          ST  CMD   COUNT
LIB1     3995-131   4    4    4  144   43   12   P 0900    Y   Y   Y  LM        0
LIB2     3995-111   4    4    3  144   48    8   P 0908    Y   Y   Y  LACT      0
LIB3     3995-132   4    4    3  144   45    6   P 0910    Y   Y   Y  LM        0
LIB4     3995-112   4    4    4  144   38    6   P 0918    Y   Y   Y  LM        4
LIB5     3995-133   4    4    0  144   11   26   P 0920    Y   Y   Y  LD       12
LIB6     3995-113   4    4    1  144   10   36   P 0928    Y   Y   Y  LM        0
LIB7     3995-C36   6    1    1  156  153    4     19F4    Y   Y   Y  LM        0
STDALONE 9246       0    0    0    0    0    0        0    N   Y                0
P3995SW3 3995-SW3   0    0    0    0    0    0        0    Y   Y                0
PCTREUS  3995-131   0    0    0    0    0    0        0    Y   Y                0
PCTWORM  3995-132   0    0    0    0    0    0        0    Y   Y                0
P3995133 3995-133   0    0    0    0    0    0        0    Y   Y                0
P3995SW4 3995-SW4   0    0    0    0    0    0        0    Y   Y                0
PSEUDO1             0    0    0    0    0    0        0    Y   Y                0
```

Displaying Storage Group Status

The following command syntax displays storage group status:

```
►►──┬─DISPLAY─┬──────────────────────────────────────────────────────►
    └─D───────┘

►─SMS,──┬─STORGRP─┬──(──┬─group-name─┬──)─,DETAIL──┬──────────────────┬──────►◄
        └─SG──────┘     └─ALL────────┘             └─,L=─┬─a──────┬─┘
                                                         ├─cc─────┤
                                                         ├─cca────┤
                                                         ├─name───┤
                                                         └─name-a─┘
```

DISPLAY
Specifies the OAM display command.

STORGRP(*group-name* | ALL)
Displays the status of an Object or Object Backup storage group. If *group-name* is specified, it displays the status of the requested storage group. If *group-name* is omitted, or if ALL is selected, it displays the status of all Object and Object Backup storage groups within the active configuration. If *group-name* is ALL, you must enclose the ALL parameter in two sets of parentheses, as follows:

```
D SMS,STORGRP((ALL))
```

DETAIL
Displays the detail status for the storage group specified by *object-group*.

L={a | cc | cca | name | name-a}
Specifies where the results of the inquiry are displayed: the display area *a*, the console *cc*, or both *cca*. The name parameter is routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

To display Object or Object Backup storage group status, enter the following command:

```
DISPLAY SMS,STORGRP(object-group),DETAIL
```

Specifying *object-group* on the D SMS,STORGRP command displays the status of the requested storage group:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP    TAPE
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2  UNIT
objgname  a bbbbb  sys_name objbusg1  objbusg2  unitname
---------------------------------------------------------
Library Names:  libname1 libname2 libname3 libname4
                libname5 libname6 libname7 libname8
                         OPTICAL        TAPE
All Volumes Full:              c           d
Writable Volumes:          eeeee       fffff
Drive Start Threshold:     ggggg       hhhhh
Volume Full Threshold:     iiiii       jjjjj
Reinitialization Mode:     kkkkk     --N/A--
# of Active Drives:         llll     mmm/nnn
```

The fields displayed in each data line are as follows:

objgname
Name of the Object storage group or Object Backup storage group being displayed.

a
Storage group type:
- **G** Group
- **B** Backup

bbbbb
Number of requests for this storage group, either waiting to be executed or currently being executed.

sys_name
Name of the system where OSMC processing is executed for the object storage group. This is blank if the storage group being displayed is an Object Backup storage group.

objbusg1
Name of the Object Backup storage group specified in the CBROAM*xx* member of PARMLIB, where the first backup copies of objects in this object storage group are stored. This is *-N/A-* if the storage group being displayed is an Object Backup storage group. This is blank if the storage group being displayed does not have a first backup storage group defined.

objbusg2
Name of the Object Backup storage group specified in the CBROAM*xx* member of PARMLIB, where the second backup copies of objects in this object storage group are stored. This is *-N/A-* if the storage group being displayed is an Object Backup storage group.

This is blank if the storage group being displayed does not have a second backup storage group defined.

unitname
: MVS esoteric or generic assigned to this storage group, specified in the CBROAM*xx* member of PARMLIB. This will be *-N/A-* if the storage group being displayed does not have a tape unit name associated with it.

libname1
: Names of the one to eight optical libraries where volumes from the optical storage group or backup optical storage group may reside. If volumes from the optical storage group reside on the shelf, then the first optical library name is the name of the pseudo library, and the other seven names are other pseudo library names or are blank.

c
: Indicates whether all optical volumes associated with this storage group are full. Valid values are “Y” or “N”.

d
: Indicates whether all tape volumes associated with this storage group are full. Valid values are “Y” or “N”.

eeeee
: Number of optical volumes, associated with this storage group, that are writable and not full.

fffff
: Number of tape volumes, associated with this storage group, that are writable and not full.

ggggg
: Optical drive startup threshold for this storage group. When the number of outstanding write requests for this storage group exceeds the number of active optical drives multiplied by the drive startup threshold, a new optical drive can be started, until all optical drives are busy.

hhhhh
: Tape drive startup threshold for this storage group. When the number of outstanding write requests for this storage group exceeds the number of active tape write drive tasks multiplied by the drive startup threshold, a new tape drive task can be started, up to the maximum number of tape store tasks for this storage group.

iiiii
: When the number of kilobytes that are free on an optical volume belonging to this storage group drops below this threshold value, the volume is marked full.

jjjjj
: When the number of kilobytes that are free on a tape volume belonging to this storage group drops below this threshold value, the volume is marked full.

kkkkk
: The optical reinitialization mode for this storage group. When a reusable optical cartridge expires, both volumes on that cartridge are scheduled to be reinitialized. At that time, the volumes can be returned to scratch, or remain assigned to their current object storage group. Valid values are “SCRATCH” or “GROUP”.

llll
: The current number of optical drives actively processing requests for this storage group.

mmm
: The current number of tape tasks actively processing writes for this storage group.

nnn
: The current number of tape tasks actively processing reads for this storage group.

The following is a sample of DISPLAY SMS,STORGRP(GROUP01),DETAIL:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP    TAPE
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2  UNIT
GROUP01   G     0  SYSTEM1  BACKUP1   BACKUP2   --------
Library Names:  P156A    PWA9
                         OPTICAL        TAPE
All Volumes Full:              -           -
Writable Volumes:              0           0
Drive Start Threshold:        10           0
Volume Full Threshold:        10           0
Reinitialization Mode:   SCRATCH     --N/A--
# of Active Drives:            0           0
```

The following is a sample of DISPLAY SMS,STORGRP(ALL),DETAIL:

```
CBR1130I OAM storage group status:
OBJECT   TY  REQ   OSMC     BACKUP    BACKUP    TAPE
STORGRP     COUNT  SYSTEM   STORGRP1  STORGRP2  UNIT
BACKUP1   B     0           --N/A--   --N/A--   --------
Library Names:  P156A    PWA9
BACKUP2   B     0           --N/A--   --N/A--   --------
Library Names:  PWA9     P156A
BKUP1     B     0           --N/A--   --N/A--   3490
Library Names:  PWA9     P156A
BKUP2     B     0           --N/A--   --N/A--   3490
Library Names:  PWA9     P156A
GROUP00   G     0 SYSTEM1   BACKUP1   BACKUP2   --------
Library Names:  P156A    PWA9
GROUP01   G     0 SYSTEM1   BACKUP1   BACKUP2   --------
Library Names:  P156A    PWA9
GROUP02   G     0 SYSTEM1   BACKUP1   BACKUP2   --------
Library Names:  P156A    PWA9
GROUP03   G     0 SYSTEM1   BACKUP1   BACKUP2   --------
Library Names:  P156A    PWA9
GROUP94   G     0           BKUP1     BKUP2     --------
Library Names:  PWA9     P156A
SBACKUP   B     0           --N/A--   --N/A--   --------
Library Names:  STDALONE PCTREUSE PCTWORM  P3995133
*NONGRP*  N     0           --N/A--   --N/A--   --------
Library Names:
*SCRTCH*  S     0           --N/A--   --N/A--   --------
Library Names:
```

Displaying Volume Status

The OAM display volume command shows if a volume belongs to an Object Backup storage group, and whether that volume is used to write first or second backup copies of objects. The first or second backup copy indicator is the backup volume type.

The following command syntax displays volume status:

```
►►──┬─DISPLAY─┬──SMS,──┬─VOLUME─┬──(volser)──┬────────────────┬──────────────►◄
    └─D───────┘        └─VOL────┘            └─,L=─┬─a──────┬─┘
                                                   ├─cc─────┤
                                                   ├─cca────┤
                                                   ├─name───┤
                                                   └─name-a─┘
```

DISPLAY
Specifies the OAM display command.

VOLUME,*volser*
: Displays the status of the requested optical volume and the optical volume on the other side of the optical disk. There is no option to display all optical disk volumes known to the system; however, you may use ISMF to display the optical volume list.

L={*a* **|** *cc* **|** *cca* **|** *name* **|** *name-a***}**
: Specifies where the results of the inquiry are to be displayed: the display area *a*, the console *cc*, or both *cca*. The *name* parameter will be routed to the console referred to by *name* and the screen referred to by *a*. The *name* parameter can be an alphanumeric character string.

Note: In order to display information in the Tape Volume table concerning objects stored on tape volumes, you must use the SPUFI SELECT command. Using the DISPLAY command against these tape volumes only provides information from the Tape Configuration database, not the Tape Volume table.

To display optical or object tape volume status, enter the following command:

```
DISPLAY SMS,VOLUME(volser)
```

Specifying *volser* displays the status of the requested optical volume and its opposite-side volume.

For an optical volume, the following variable information is displayed in message CBR1140I:

```
CBR1140I  OAM volume status
VOLUME  STORAGE RD  WR  WP MEDIA   FREE SPACE     MOUNTED   PENDING  REQ
        GROUP              TYPE    (KB)      (%)  DRIVE     MOUNT    CT
volser  sgname  a   b   c  mediatyp freespac fff% mdrvname pdrvname  ggg
oppvol  sgname  a   b   c  mediatyp freespac fff% mdrvname pdrvname  ggg
media descript {WORM|rewritable|unknown} optical disk media.
LIBRARY: libname
{SHELF LOC: shelfloc | PSEUDO LIBRARY: plibname}
OWNER: owner information
XCF MEMBER NAME: member-name
BACKUP TYPE: {BACKUP1 | BACKUP2}
CREATION DATE: createdate     ENTER-EJECT DATE: eedate
VOLSER:             volser     oppvol
LAST WRITTEN DATE: lwdate     lwdate
LAST MOUNTED DATE: lmdate     lmdate
EXPIRATION DATE:   expdate    expdate
```

The fields displayed in each data line are as follows:

volser
: Volume serial number of the requested optical volume.

oppvol
: Volume serial number of the opposite side of the requested optical volume.

sgname
: Name of the storage group to which the optical volume belongs.

a
: Optical volume label readability status:
  - **Y** Readable
  - **N** Unreadable

b
: Optical volume writability status:

Y Writable
N Unwritable

c
Optical volume write protection status:
Y Write protected
N Not write protected

mediatyp
8-character or less media type of the requested optical volume. Valid values are as follows:

9247	LMSI (2000 MB) 12-inch, WORM optical disk media
3995-1RW	3995 5.25-inch, single-density (650 MB), rewritable optical disk media
3995-1WO	3995 5.25-inch, single-density (650 MB), WORM optical disk media
3995-2RW	3995 5.25-inch, double-density (1300 MB), rewritable optical disk media
3995-2WO	3995 5.25-inch, double-density (1300 MB), WORM optical disk media
3995-4RW	3995 5.25-inch, quad-density (2600 MB) rewritable optical disk media
3995-4WO	3995 5.25-inch, quad-density (2600 MB) WORM optical disk media
3995-8RW	3995 5.25-inch, 8x-density (5.2 GB) rewritable optical disk media
3995-8WO	3995 5.25-inch, 8x-density (5.2 GB) WORM optical disk media

Note: Double-, quad-, and 8x-density WORM includes CCW media. CCW is continuous composite WORM media. WORM is write-once, read-many media.

freespac
Remaining volume space of the requested optical volume in kilobytes (KB).

fff%
Percentage of free space on the optical volume. For a full optical volume, this field contains “FULL”.

Note: With the release of DFSMS/MVS Version 1 Release 2, the percentage of free space is based on total volume capacity. This may result in a slight difference in free space percentage from prior releases.

mdrvname
Name of the drive where this optical volume is mounted. If the optical volume is not mounted, this field contains blanks.

pdrvname
For a 9247 drive, this field displays the name of the drive where a mount is pending for this optical volume. If no mount is pending, this field contains blanks.

For a 3995 drive, this field displays “YES” if a mount is pending for this optical volume.

ggg
Number of read requests for this optical volume that are currently pending in OAM.

optical media description
72-character description for the requested optical volume.

libname
Name of the library in which the optical volume resides. This field appears only for a library optical volume.

shelfloc Shelf location of a volume after it has been ejected from the library.

plibname The name of the pseudo library that the volume is associated with. This field appears only for a shelf optical volume.

owner-information
Owner information from the optical volume label.

member-name The XCF member name of the OAM that is currently managing and controlling the optical volume.

BACKUP1 **|** *BACKUP2*
The backup volume type for this volume. This line is only displayed if the volume type is "B" (a backup volume). This volume contains first (BACKUP1) or second (BACKUP2) backup copies written to the Object Backup storage group to which this volume belongs.

volser The volume serial number of the requested optical volume.

oppvol The volume serial number of the opposite side of the requested optical volume.

createdate Date the optical volume was created in the format YYYY-MM-DD.

lwdate Date the optical volume was last written to in the format YYYY-MM-DD.

lmdate Date the optical volume was last mounted in the format YYYY-MM-DD.

eedate Date the optical volume was last entered or ejected from the library in the format YYYY-MM-DD.

expdate Expiration date of the optical volume in the format YYYY-MM-DD.

status Additional optical volume status:
- Optical volumes are not in their assigned optical library slot.
- The optical library slot assigned to these optical volumes is empty or contains different optical volumes.

For an optical volume belonging to an Object storage group, the command `D SMS,VOLUME(WG830B)` displays the following information:

```
CBR1140I OAM volume status:
VOLUME STORAGE  RD WR WP MEDIA    FREE SPACE    MOUNTED  PENDING   REQ
       GROUP             TYPE       (KB)   (%)  DRIVE    MOUNT     CT
WG830B GROUP22  Y  Y  N  3995-1WO   194044  65% P9D3     --------    0
WG830A GROUP22  Y  Y  N  3995-1WO   194656  65%          --------    0
3995 (650 MB) WORM optical disk media.
LIBRARY: PWA9
PSEUDO LIBRARY:
OWNER: E. M. WHALEN
XCF MEMBER NAME: -N/A-
CREATION DATE: 2000-12-19  ENTER-EJECT DATE:  2000-12-19
VOLSER:           - WG830B -  - WG830A -
LAST WRITTEN DATE: 2001-03-08  2000-12-19
LAST MOUNTED DATE: 2001-03-08  2000-12-19
EXPIRATION DATE:   0001-01-01  0001-01-01
```

For an optical volume belonging to an Object Backup storage group, the command `D SMS,VOLUME(WG920A)` displays the following information:

```
CBR1140I OAM volume status:
VOLUME STORAGE  RD WR WP MEDIA    FREE SPACE    MOUNTED  PENDING   REQ
       GROUP             TYPE       (KB)   (%)  DRIVE    MOUNT     CT
WG920A BACKUP1  Y  Y  N  3995-1WO    60838  20%          --------    0
WG920B BACKUP1  Y  Y  N  3995-1WO   166181  55%          --------    0
3995 (650 MB) WORM optical disk media.
LIBRARY: PWA9
PSEUDO LIBRARY:
OWNER: E. M. WHALEN
XCF MEMBER NAME: -N/A-
BACKUP TYPE: BACKUP1
CREATION DATE: 2000-12-07  ENTER-EJECT DATE:  2000-12-07
VOLSER:            - WG920A -  - WG920B -
LAST WRITTEN DATE: 2001-01-08  2000-12-07
LAST MOUNTED DATE: 2001-02-14  2000-12-07
EXPIRATION DATE:   9999-12-31  0001-01-01
```

For an object tape volume, the following information is displayed:

```
CBR1240I  OAM object tape volume status
VOLUME  STORAGE UNITNAME RD  WR  CM  IN   MED   FREE-SPACE   %    LOST  REQ
        GROUP                        USE  TYPE              FULL  FLAG  CT
volser  sgname  unitname a   b   c    d   ee   ffffffffff    gg    h   iii
XCF MEMBER NAME: member-name
BACKUP TYPE: {BACKUP1 | BACKUP2}
CAPACITY:          capacity     ERDS PHYSICAL ID:  epi
CREATION DATE:     createdate   EXPIRATION DATE:   expdate
LAST MOUNTED DATE: lmdate       LAST WRITTEN DATE: lwdate
```

The fields displayed in each data line are as follows:

volser Volume serial number of the requested tape volume.

sgname Name of the object storage group to which the tape volume belongs.

unitname MVS unit name used when the tape volume is allocated. If the tape volume is in an IBM tape library, this value is ignored.

a Tape volume readability status:
- **Y** Readable
- **N** Unreadable

b Tape volume writability status:
- **Y** Writable
- **N** Unwritable

c Compaction indicator for this tape volume:
- **Y** Tape volume written in compacted format.
- **N** Tape volume written in noncompacted format.

d Tape volume in use indicator for this tape volume:
- **Y** Tape volume currently in use by an OAM drive task.
- **N** Tape volume not currently in use by an OAM drive task.

ee Media type of the requested tape volume:
- **02** IBM Cartridge System Tape
- **04** IBM Enhanced Capacity Cartridge System Tape
- **05** IBM High Performance Cartridge System Tape
- **06** Extended High Performance Cartridge System Tape

ffffffffff Remaining space on the requested tape volume in kilobytes (KB).

gg Percentage that the requested tape volume is full (percentage of the tape that has been used).

h Volume lost indicator.

iii Number of read requests for this tape volume which are currently pending in OAM.

member-name The XCF member name of the OAM which is currently managing and controlling this tape volume.

BACKUP1 **|** *BACKUP2*
The backup volume type for this volume. This volume contains first (BACKUP1) or second (BACKUP2) backup copies written to the Object Backup storage group to which this volume belongs.

capacity Approximate number of millimeters of tape or approximate number of kilobytes of data which can be written to the volume, allowing variance for different manufacturers.

epi The ERDS Physical Identifier (EPI) which indicates the real underlying device type that is used to write OAM objects to this volume. This is used to assist in problem diagnosis in a mixed device environment where native and emulated devices coexist.

createdate Date the tape volume was created, in the format YYYY-MM-DD.

expdate Expiration date of the tape volume, in the format YYYY-MM-DD.

lmdate Date the tape volume was last mounted, in the format YYYY-MM-DD.

lwdate Date the tape volume was last written to, in the format YYYY-MM-DD.

For an object tape volume belonging to an Object storage group, the command D SMS,VOLUME(TNN000) displays the following information:

```
CBR1240I Object tape vol status:
VOLUME STORAGE  UNITNAME RD WR CM IN  MED  FREE-SPACE  %   LOST REQ
       GROUP                      USE TYPE            FULL FLAG CT
TNN000 GROUP24  3490      Y  Y  Y  N   04      874212   0   N     0
XCF MEMBER NAME: -N/A-
CAPACITY:             874218
ERDS PHYSICAL ID:       0002
CREATION DATE: 2000-12-18    EXPIRATION DATE: 9999-12-31
LAST MOUNTED DATE: 2001-02-23    LAST WRITTEN DATE:  2000-12-18
```

For an object tape volume belonging to an Object Backup storage group, the command D SMS,VOLUME(TNN005) displays the following information:

```
CBR1240I Object tape vol status:
VOLUME STORAGE  UNITNAME RD WR CM IN  MED  FREE-SPACE  %   LOST REQ
       GROUP                      USE TYPE            FULL FLAG CT
TNN005 BACKUP2  3490      Y  Y  Y  N   04      874210   0   N     0
XCF MEMBER NAME: -N/A-
BACKUP TYPE: BACKUP2
CAPACITY:             874218
ERDS PHYSICAL ID:       0002
CREATION DATE: 2000-12-18    EXPIRATION DATE: 9999-12-31
LAST MOUNTED DATE: 2001-03-08    LAST WRITTEN DATE:  2000-12-18
```

Note: If the object tape volume resides in an IBM automated tape library, you will receive both the CBR1240I display for object tape volumes and the CBR1180I display for tape library volumes. Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for an example of CBR1180I.

Displaying Volumes that Have LOSTFLAG Set

The OAM display lost volumes command displays the volume serial numbers of the optical and tape volumes that have LOSTFLAG set. OAM requests may fail if this flag is set; this command allows you to determine which volumes are “lost” and to perform the necessary troubleshooting on those volumes.

The following command syntax displays the optical and tape volumes that have LOSTFLAG set:

```
►►──┬─MODIFY─┬──OAM,──┬─DISPLAY,─┬──LOSTVOL──────────────────────────────────────►◄
    └─F──────┘        └─D,───────┘
```

The following is a description of the keyword for this command:

LOSTVOL
Displays the optical and tape volumes that have LOSTFLAG set.

To display the optical and tape volumes that have LOSTFLAG set, enter the following command:

```
F OAM,DISPLAY,LOSTVOL
```

The output of the F OAM,DISPLAY,LOSTVOL command looks as follows if there are optical volumes that are lost, but no tape volumes that are lost:

```
F OAM,DISPLAY,LOSTVOL
CBR1154I OPTICAL volume OPTVOL1 is a lost volume.
CBR1154I OPTICAL volume OPTVOL2 is a lost volume.
CBR1155I Total number of OPTICAL volumes marked lost is 2.
CBR1155I Total number of TAPE volumes marked lost is 0.
```

The output of the F OAM,DISPLAY,LOSTVOL command looks as follows if there are both optical and object tape volumes that are lost:

```
F OAM,DISPLAY,LOSTVOL
CBR1154I OPTICAL volume OPTVOL1 is a lost volume.
CBR1154I OPTICAL volume OPTVOL2 is a lost volume.
CBR1155I Total number of OPTICAL volumes marked lost is 2.
CBR1154I TAPE volume TAPVOL1 is a lost volume.
CBR1155I Total number of TAPE volumes marked lost is 1.
```

Displaying SETOPT, SETOAM, and SETOSMC Statements

The F OAM,DISPLAY,SETOAM, SETOPT, or SETOSMC command is used to display the current settings of the SETOAM, SETOPT, or SETOSMC statement for the OAM address space.

The following is the command syntax:

```
►►──┬─MODIFY─┬──OAM,DISPLAY,──┬─SETOAM──┬─,scope──────────────────────────────────►◄
    └─F──────┘                ├─SETOPT──┤
                              └─SETOSMC─┘
```

The following are the descriptions of the keywords used in this command:

SETOPT | SETOAM | SETOSMC
Specifies the command parameter being displayed. SETOAM is used to display values of settings in an object tape environment. SETOPT is used to display values of settings in an optical environment. SETOSMC is used to display the values of settings for OSMC processing.

scope
Specifies what information will be displayed. For the SETOAM and SETOPT parameters, the valid values are:

ALL Displays the settings for each valid storage group as well as the global default settings. If ALL is specified, then the global default, if applicable, is displayed as well as the settings for each valid storage group in the active SMS configuration.

storgrp
Displays only the settings for the specified storage group name. There can be up to fourteen storage group names indicated on a single DISPLAY command.

For the SETOSMC parameter, the valid values are:

storgrp
Displays the settings for the FIRSTBACKUPGROUP and SECONDBACKUPGROUP for the specified storage group name.

The following is a sample of DISPLAY SMS,SETOAM,*storgrp* command:

```
F OAM,DISPLAY,SETOAM,GROUP22

CBR1075I GROUP22 value for EXPDATE is 2019/031
CBR1075I GROUP22 value for TFULLTHR is 240
CBR1075I GROUP22 value for TFULLPER is 100
CBR1075I GROUP22 value for TAPEUNIT is 3490
CBR1075I GROUP22 value for DMWT is 120
CBR1075I GROUP22 value for DATACL is ATLM2CU
CBR1075I GROUP22 value for TCOMP is N
CBR1075I GROUP22 value for TDRVSTRT is 9999
```

The following is a sample of DISPLAY SMS,SETOAM,ALL command:

```
F OAM,DISPLAY,SETOAM,ALL

CBR1075I GLOBAL value for EXPDATE is 2019/031
CBR1075I GLOBAL value for TFULLTHR is 240
CBR1075I GLOBAL value for MWT is 5
CBR1075I GLOBAL value for DATACL is ATLM2CU
CBR1075I GROUP00 value for TFULLPER is 0
CBR1075I GROUP00 value for TAPEUNIT is 3490
CBR1075I GROUP00 value for DMWT is 0
CBR1075I GROUP00 value for TCOMP is N
CBR1075I GROUP00 value for TDRVSTRT is 0
CBR1075I GROUP00 value for TFULLPER is 100
CBR1075I GROUP22 value for TAPEUNIT is 3490
CBR1075I GROUP22 value for DMWT is 120
CBR1075I GROUP22 value for TCOMP is N
CBR1075I GROUP22 value for TDRVSTRT is 9999
```

The following is a sample of DISPLAY SMS,SETOPT,*storgrp* command:

```
F OAM,DISPLAY,SETOPT,GROUP22

CBR1075I GROUP22 value for OPREINIT is GROUP
CBR1075I GROUP22 value for OPDISDLY is 0
```

The following is a sample of DISPLAY SMS,SETOPT,ALL command:

```
F OAM,DISPLAY,SETOPT,ALL

CBR1075I GLOBAL value for OPREINIT is GROUP
CBR1075I GLOBAL value for OPDISDLY is 0
CBR1075I GLOBAL value for MWT is 3
CBR1075I GROUP00 value for OPREINIT is GROUP
CBR1075I GROUP22 value for OPREINIT is GROUP
CBR1075I GROUP26 value for OPREINIT is GROUP
CBR1075I GROUP28 value for OPREINIT is GROUP
```

The following is a sample of DISPLAY SMS,SETOSMC,*storgrp* command:

```
F OAM,DISPLAY,SETOSMC,GROUP28

CBR1075I GROUP28 value for FIRSTBACKUPGROUP is backup1
CBR1075I GROUP28 value for SECONDBACKUPGROUP is backup2
```

The following is a sample of DISPLAY SMS,SETOSMC,ALL command:

```
F OAM,DISPLAY,SETOSMC,ALL

CBR1075I GLOBAL value for FIRSTBACKUPGROUP is backup1
CBR1075I GLOBAL value for SECONDBACKUPGROUP is backup2
```

Displaying Outstanding OAM Messages

To display outstanding OAM messages, enter the following command:

```
DISPLAY R,L,KEY=OAM
```

Use this command to display the message identification numbers and texts of all immediate action, eventual action messages, and messages waiting for replies that OAM issued.

Querying Summary and Detail Information for Pending and Active Requests

The F OAM,QUERY,*options* command is used to display:

- A summary of active optical requests
- A summary of active tape requests
- A summary of waiting optical requests
- A summary of waiting tape requests
- Detailed information concerning active optical requests
- Detailed information concerning active tape requests
- Detailed information concerning waiting optical requests
- Detailed information concerning waiting object tape requests

The following command syntax shows the QUERY command:

```
                                         ┌─SUMMARY,─┐
►►──┬─MODIFY─┬──OAM,──QUERY,──┬─ACTIVE,──┬──┼─DETAIL,──┼──┬─ALL────┬──────────►◄
    └─F──────┘                └─WAITING,─┘  └─SUMMARY,─┘  ├─READ───┤
                                                          ├─WRITE──┤
                                                          ├─DELETE─┤
                                                          ├─EJECT──┤
                                                          ├─ENTRY──┤
                                                          ├─AUDIT──┤
                                                          ├─LABEL──┤
                                                          └─REMAP──┘
```

Note: OAM is the default name of the cataloged procedure in your SYS1.PROCLIB. If a name other than OAM is used for the cataloged procedure, use that name in the QUERY statement. For example, MODIFY *procname_name*,QUERY,ACTIVE,SUMMARY.

The following are the keyword descriptions of the QUERY command:

QUERY
Specifies a request to display information about active and waiting tape and optical requests. The abbreviation for this command, Q, may also be used.

ACTIVE
Indicates that only information about active requests, those currently being processed, is to be displayed. The abbreviation for this command, A, may also be used. Either the ACTIVE or WAITING keyword must be specified on the QUERY statement.

WAITING
Indicates that only information about requests waiting to be processed are to be displayed. The abbreviation for this command, W, may also be used. Either the WAITING or ACTIVE keyword

must be specified on the QUERY command. This includes any currently waiting requests that have been sent to other instances of OAM in the OAMplex for processing.

SUMMARY
Indicates that only summary information about the requested category is to be displayed. If neither the SUMMARY nor the DETAIL keyword is specified on the QUERY command, then only summary information is displayed for the requested category. The abbreviation for this command, S, may also be used. This summary information also includes any currently waiting requests that have been sent to other instances of OAM in the OAMplex for processing. This is the default.

DETAIL
Indicates that only detailed information about the requested category (ACTIVE or WAITING) is to be displayed. If neither the DETAIL nor the SUMMARY keyword is specified on the QUERY command, then only summary information is displayed for the requested category. The abbreviation for this command, D, may also be used. Query detail messages are written to the hard copy log so that the WTO buffers are not overrun, which causes a system degradation. One of the following keywords is required when the DETAIL keyword is specified:

ALL
Indicates that detail information for either all active or all waiting (depending on the prior specification in the command) requests are to be displayed.

READ
Indicates that detail information for all READS (active or waiting) is to be displayed.

WRITE
Indicates that detail information for either all WRITES (active or waiting) is to be displayed.

DELETE
Indicates that detail information for all DELETES (active or waiting) is to be displayed.

EJECT
Indicates that detail information for all EJECTS (active or waiting) is to be displayed.

ENTRY
Indicates that detail information for all ENTRIES (active or waiting) is to be displayed.

AUDIT
Indicates that detail information for all AUDITS (active or waiting) is to be displayed.

LABEL
Indicates that detail information for all LABELS (active or waiting) is to be displayed.

REMAP
Indicates that detail information for all REMAPS (active or waiting) is to be displayed.

Note: The OAM QUERY command is passed to the OAM address space through the MVS MODIFY system command. All messages sent as a result of the OAM QUERY command are sent to the MVS/ESA console from which the command originated, with the exception of the QUERY DETAIL messages. They are only sent to the system log, not the console. The system operator should be aware that entering a F OAM,QUERY,ACTIVE,DETAIL or a F OAM,QUERY,WAITING,DETAIL command can result in a significant number of messages being issued by OAM if there is a significant backlog of OAM requests processing or waiting for execution. Refer to *z/OS MVS System Messages, Vol 4* for further information concerning messages associated with the OAM QUERY command.

To query OAM to provide information about active optical and tape requests, enter the following command:

```
F OAM,QUERY,ACTIVE

      OR

F OAM,QUERY,ACTIVE,SUMMARY
```

The following information is displayed only if optical libraries are defined in the active configuration:

```
CBR1720I OPTICAL ACTIVE SUMMARY
----------- OPTICAL REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES  DELETES  ENTERS  EJECTS  AUDITS  LABELS
aaaaaa   bbbbbb  cccccc  dddddd  eeeeee  ffffff  gggggg
```

aaaaaa
: Total number of object read requests from an optical volume currently being processed. This includes read requests being process on this system that originated from another instance of OAM in an OAMplex.

bbbbbb
: Total number of object write requests to an optical volume currently being processed. This includes write requests being processed on this system that originated from another instance of OAM in an OAMplex.

cccccc
: Total number of object delete requests from an optical volume currently being processed.

dddddd
: Total number of optical volume enter requests currently being processed.

eeeeee
: Total number of optical volume eject requests currently being processed. This number also includes system initiated ejects.

ffffff
: Total number of optical volume audit requests currently being processed.

gggggg
: Total number of optical cartridge label requests currently being processed.

The following information is displayed for object tape requests only if there are SETOAM statements in the current OAM invocation:

```
CBR1720I OPTICAL ACTIVE SUMMARY
----------- OPTICAL REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES  DELETES  ENTERS  EJECTS  AUDITS  LABELS
aaaaaa   bbbbbb  cccccc  dddddd  eeeeee  ffffff  gggggg
```

The following fields in the data line specify the number of each resource currently being processed:

aaaaaa
: Total number of object read requests from a tape volume currently being processed. This includes read requests being processed on this system that originated from another instance of OAM in an OAMplex.

bbbbbb
: Total number of object write requests to a tape volume currently being processed.

To query OAM to provide information about waiting optical and tape requests, enter the following command:

```
F OAM,QUERY,WAITING

      OR

F OAM,QUERY,WAITING,SUMMARY
```

The following information is displayed only if optical libraries are defined in the active configuration:

```
CBR1730I TAPE OBJECT ACTIVE SUMMARY:

----------- OBJECT TAPE REQUESTS CURRENTLY BEING PROCESSED--------------
 READS  WRITES
aaaaaa   bbbbbb
```

The following fields in the data line specify the number of each resource waiting for execution:

aaaaaa
: Total number of object read requests from an optical volume waiting to be processed. This includes read requests waiting to be processed on this system that originated from another instance of OAM in an OAMplex or read requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

bbbbbb
: Total number of object write requests to an optical volume waiting to be processed. This includes write requests waiting to be processed on this system that originated from another instance of OAM in an OAMplex or write requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

cccccc
: Total number of object delete requests from an optical volume waiting to be processed.

dddddd
: Total number of optical volume enter requests waiting to be processed.

eeeeee
: Total number of optical volume eject requests waiting to be processed. This number also includes system initiated ejected.

ffffff
: Total number of optical volume audit requests waiting to be processed.

gggggg
: Total number of optical cartridge label requests waiting to be processed.

The following information is also displayed for object tape requests only if there are SETOAM statements in the current OAM invocation:

```
CBR1700I OPTICAL WAITING SUMMARY:

------------- OPTICAL REQUESTS WAITING FOR PROCESSING---------------------
READS    WRITES    DELETES   ENTERS    EJECTS    AUDITS    LABELS
aaaaaa   bbbbbb    cccccc    dddddd    eeeeee   ffffff     gggggg
```

The following fields in the data line specify the number of each resource waiting for execution:

aaaaaa Total number of object read requests from a tape volume waiting to be processed. This includes read requests waiting to be processed on this system that originated from another instance of OAM in an OAMplex or read requests originated by this system, waiting to be processed by another instance of OAM in the OAMplex.

bbbbbb Total number of object write requests to a tape volume waiting to be processed.

Scheduling an SVC Dump for the OAM Address Space

OAM uses SVC dumps as a diagnostic tool for system hangs or performance problems. In order to capture this data, the operator must issue the DUMP command after the completion of a recreate to obtain all the required data needed for diagnostics. OAM provides a streamlined version of the DUMP command. The F OAM,DUMP,(*operands*) command automatically collects all the pertinent data needed for diagnostic purposes without the operator having to key in all the correct parameters.

The following command syntax shows the DUMP command:

```
►►──┬─MODIFY─┬──OAM,──DUMP,──┬─OAM────────────────────────┬──────────────────►◄
    └─F──────┘               ├─ALL────────────────────────┤
                             ├─ASID,asid1,asid2,...───────┤
                             └─JOBN,jobname1,jobname2,...─┘
```

OAM Specifies an SVC dump is scheduled for the OAM address space. If the first operand after the DUMP verb is either OAM or blank, OAM schedules a SVC dump for the OAM address space.

ALL An SVC dump is scheduled for the OAM address space and any address spaces which currently have work queued to the OAM address space, up to 14 address spaces in addition to OAM.

If the first operand after the DUMP verb is ALL, OAM scans all queues to identify address spaces that are not the OAM address space. OAM scans until all queues are searched or 14 address are found. OAM then schedules an SVC dump for the OAM address space and up to 14 other address spaces that currently have work queued in the OAM address space.

ASID,*asid1, asid2, asid3...*
An SVC dump is scheduled for the OAM address space and any address spaces separated by commas specified after the ASID operand. A valid ASID is a 1 to 4 hexadecimal (0–9, A–F) value. From one to 14 ASIDs can be specified with the ASID operand. If more than 14 ASIDs are specified, the first 14 will be used.

If the first operand after the DUMP verb is ASID, OAM validates that any ASIDs specified following the ASID operand are valid hexadecimal characters (0–9, A–F). If they are valid, OAM, schedules an SVC dump for the OAM address space and any additional address spaces specified (up to 14 address spaces in addition to OAM).

JOBN,*jobname1,jobname2,jobname3...*
An SVC dump is scheduled for the OAM address space and any job spaces specified after the JOBN operand separated by commas. A valid job name is a 1 to 8 character value of the following character set:
- Alphanumeric characters (A–Z, 0–9)
- National characters (&, $, @)
- Wildcard characters (*, ?) where '*' can stand for 0 or more characters, up to the maximum length of the job name string (8) and '?' can stand for one character.

From 1 to 14 job names can be specified with the JOBN operand. If more than 14 job names are specified, the first 14 will be used.

If the first operand after the DUMP verb is JOBN, OAM validates that any job names specified following the JOBN operand contain the valid character set. If they are valid, OAM schedules an SVC dump for the OAM address space and any job names specified (up to 14 jobs in addition to OAM).

OAM issues messages for any errors found in the DUMP command at SVC scheduling time and at SVC DUMP data capture completion. For more information concerning these messages, refer to *z/OS MVS System Messages, Vol 4*.

Restarting the OAM Address Space

The OAM RESTART command causes the OAM address space to perform restart processing. During restart processing, OAM matches the constructs and definitions used to those that are found in the active SMS configuration.

This command provides the ability to avoid having to do a STOP and START of the OAM address space and allows the OAM address space to retain its current ASID.

To restart the OAM address space without first stopping OAM, enter the following RESTART command:

```
F OAM,RESTART,parm=value,parm=value,...
```

This command can be issued when the parameter RESTART=NO is specified on the OAM procedure statement, which prevents OAM from automatically restarting the address space after a change is made to the SMS configuration. The time that OAM is notified of the SCDS activation depends on the time interval specified with the INTERVAL keyword in the IGDSMS*xx* PARMLIB member. If the change to the SMS configuration affects OAM and a restart of the OAM address space is required, you can issue this command in place of issuing a STOP and START command.

The following command syntax shows the RESTART command. Use the OAM parameter to specify which CBROAM*xx* PARMLIB member is used during initialization.

```
►►──┬─MODIFY─┬──OAM,──RESTART,──┬─────────────────────┬──────────────────────────►◄
    └─F──────┘                  ├─────────────────────┤
                                │ └─OAM──=──xx─┘      │
                                ├─────────────────────┤
                                │ └─OSM──=──┬─YES──┬┘ │
                                │           ├─NO───┤  │
                                │           └─FAST─┘  │
                                ├─────────────────────┤
                                │ └─RES──=──┬─YES─┬┘  │
                                │           └─NO──┘   │
                                ├─────────────────────┤
                                │ └─MAXS──=──nn─┘     │
                                ├─────────────────────┤
                                │ └─EJECT──=──┬─W─┬┘  │
                                │             └─M─┘   │
                                └─────────────────────┘
                                  └─UNL──=──nnnn─┘
```

OAM OAM=*xx* specifies the suffix of the CBROAM*xx* PARMLIB member that OAM should process during OAM address space initialization. The two alphanumeric characters (*xx*) must immediately follow the OAM=*keyword* in the PARM field. If the two characters immediately following the OAM=*keyword* are invalid or not specified, error message CBR0025I is issued. OAM only reads PARMLIB member CBROAM*xx* if the OAM=*keyword* is specified on the PARM field of the JCL EXEC statement in the OAM cataloged procedure. If no OAM=*keyword* is specified on the PARM field of the JCL EXEC statement, no PARMLIB member is read by OAM and object tape support is not active. If the object tape support is not active, OAM cannot read any objects back or write any new objects to tape until OAM is initialized with a valid OAM=*xx* specification, and a valid corresponding CBROAM*xx* PARMLIB member. OAM processes PARMLIB member CBROAM*xx* during OAM address space initialization.

OSM Indicates whether or not OSMC is initialized at the same time as OAM after the RESTART command has been issued. The valid values for this parameter are as follows:

- **YES** OAM initializes with OSMC.
- **NO** OAM initializes without OSMC
- **FAST** OAM initializes with OSMC. However, OAM bypasses the collection audit, delaying it until OSMC storage group processing is done.

RES Determines whether OAM should automatically restart when it receives notification that a new SCDS is activated. The valid values for this parameter are as follows:

- **YES** OAM automatically restarts when a new SCDS is activated.
- **NO** OAM does not automatically restart when a new SCDS is activated.

MAXS MAXS=*nn* specifies the maximum number of OSMC storage management tasks that can be active at one time, where *nn* is a number value 1–99. The value given for MAXS must not exceed the number of optical drives that is available for storage management processing. If the MAXS parameter is not specified, a default of 2 is assigned. See “OAM Cataloged Procedure Parameter (MAXS)” on page 135 for more information about the MAXS parameter.

Note: If concurrent processing includes Object storage groups writing to tape volumes, the correct corresponding (global level) MAXTAPERETRIEVETASKS and MAXTAPESTORETASKS values on the SETOAM statement must be specified. For more information concerning these keywords, see MAXTAPERETRIEVETASKS on page 94, and MAXTAPESTORETASKS on page 94.

EJECT
Used to determine which volumes are ejected from an optical library when the library is full and there is a request to add additional volumes to the library. The valid values for this parameter are as follows:

W This value indicates that Library Space Management is to use the *least recently written* algorithm when selecting a volume to be ejected from an optical library.

M This value indicates that Library Space Management is to use the *least recently mounted* algorithm when selecting a volume to be ejected from an optical library.

UNL UNL=*nnnn* specifies the number of seconds of inactivity that OAM waits before unloading an optical drive in a library in order to keep at least one drive empty. This unload only occurs if there are no available optical drives within this library. That is, there are no empty online and operational drives. Thus, during periods of inactivity, you can cause at least one drive to be ready to accept the next mount request without first having to do a demount.

Using the UPDATE Command to Set SETOAM and SETOPT Values

Most of the current values of the SETOAM and SETOPT statements can be changed using the F OAM,UPDATE command without having to restart OAM. This command can also define settings of most of the SETOAM and SETOPT values if a CBROAM*xx* PARMLIB member was not used at OAM startup. This command provides dynamic processing that provides another method (other than using the CBROAM*xx* PARMLIB member) for customizing your object tape and optical support.

If a CBROAM*xx* PARMLIB member was used at OAM initialization, the OAM UPDATE command can be used to change the current settings for SETOAM and SETOPT for the duration of the OAM session. If OAM is restarted, in most cases, the settings from the CBROAM*xx* PARMLIB member override any SETOAM or SETOPT values modified by an update command. There is an exception to this instance. If a storage group is not explicitly specified in the CBROAM*xx* PARMLIB member, no values are stored for the storage group during OAM restart (initialization) processing since there are no defaults at the storage group level for the SETOAM and SETOPT statements. Therefore, if OAM is restarted and no specific statements exist for a storage group, and a previous UPDATE SETOPT or SETOAM command was issued with a scope of ALL to update a field, the value assigned by the update is retained. The following is an example of this scenario:

- The F OAM,UPDATE,SETOAM,ALL,DMWT,5 command is issued.
- The OAM address space is restarted.
- There are no statements for storage group GROUP00 in the CBROAM*xx* PARMLIB member.
- When the OAM restart is complete, GROUP00 will still have the updated demount wait time value of 5.
- If OAM is stopped and then started, previously set values are not retained.

The following is the syntax for the F OAM,UPDATE,SETOAM | SETOPT command:

```
►►──┬─MODIFY─┬──OAM,UPDATE,──┬─SETOAM─┬─,scope,field,value,field,value──────────►◄
    └─F──────┘               └─SETOPT─┘
```

Note: There is a maximum of seven pairs of field and value updates allowed on one UPDATE command.

The following are the descriptions of the keywords used in this command:

SETOPT | SETOAM
Specifies the command parameter being updated. SETOAM is used to update settings values in an object tape environment. SETOPT is used to update settings values in an optical environment.

scope Specifies which storage groups will be affected by the update. Valid values are ALL or the name of a storage group defined in the active SMS configuration. If ALL is specified, then the global default, if applicable, is updated for each object storage group in the active SMS configuration. If the name of a storage group is specified, only that storage group is updated with the setting value changes. Only one storage group can be specified with a single UPDATE command.

field Specifies specific keyword in the SETOAM or SETOPT statement that is being updated. Table 38 lists the valid UPDATE command keywords and their associated SETOAM keywords. If a SETOAM keyword is not on the list, it is not modifiable on the F OAM,UPDATE command.

Table 38. Valid SETOAM Keywords on the UPDATE Command

UPDATE Keyword	Associated SETOAM Keyword
EXPDATE	TAPEEXPIRATION
TFULLTHR	TAPEFULLTHRESHOLD
TFULLPER	TAPEPERCENTFULL
TAPEUNIT	TAPEUNITNAME
DMWT	DEMOUNTWAITTIME
MWT	MOUNTWAITTIME
DATACL	DATACLASS
TCOMP	TAPECOMPACTION / NOTAPECOMPACTION
TDRVSTRT	TAPEDRIVESTARTUP

Table 39 lists the valid UPDATE command keywords and their associated SETOPT keywords. If a SETOPT keyword is not on the list, it is not modifiable on the F OAM,UPDATE command.

Table 39. Valid SETOPT Keywords on the UPDATE Command

UPDATE Keyword	Associated SETOPT Keyword
OPREINIT	OPTICALREINITMODE
OPDISDLY	OPTICALDISPATCHERDELAY
MWT	MOUNTWAITTIME

value Specifies the new value for the specified field. This value must conform to the same conditions and restrictions that apply to the CBROAM*xx* PARMLIB member in SYS1.PARMLIB, with two exceptions:

- The OPREINIT (OPTICALREINITMODE) keyword values are GROUP and OAMSCR when issued with the UPDATE command. This is due to an 8-character restriction on the input values.
- The TCOMP (TAPECOMPACTION / NOTAPECOMPACTION) keyword values are Y (indicating TAPECOMPACTION) or N (indicating NOTAPECOMPACTION) when issued with the UPDATE command.

The following are some examples of these various UPDATE commands and explanation of the associated modifications:

```
F OAM,UPDATE,SETOAM,GROUP22,TAPEUNIT,TAPEESO
```

Assuming TAPEESO is a valid tape esoteric name defined to the system, this sets the TAPEUNITNAME setting for only GROUP22 to the tape esoteric, TAPEESO.

Note: TAPEUNITNAME is the MVS unit name that OAM uses to initially allocate a scratch tape when an object is stored to this Object or Object Backup storage group and stored on a tape volume. If there are tape volumes already belonging to the object or object backup storage group, they will be used first before allocating a scratch volume. Even though there is a tape unit name specified for the group, the ACS routines (for environment ALLOC) can override the TAPEUNITNAME specification by assigning the allocation to a Tape storage group, thereby, steering the allocation into an ATLDS or an MTL.

```
F OAM,UPDATE,SETOAM,ALL,TFULLPER,90,TFULLTHR,100
```

This updates all global values and every storage group name in the active SMS configuration to a TAPEPERCENTFULL value of 90 and a TAPEFULLTHRESHOLD value of 100.

```
F OAM,UPDATE,SETOAM,GROUP22,MWT,2,DMWT,4,TFULLPER,95
```

This updates the values for MOUNTWAITTIME, DEMOUNTWAITTIME, and TAPEFULLPERCENT for only the GROUP22 storage group.

```
F OAM,UPDATE,SETOPT,ALL,OPREINIT,MWT,3,OAMSCR,OPDISDLY,50
```

This updates the values for OPTICALREINITMODE, MOUNTWAITTIME, and OPTICALDISPATCHERDELAY for all global values and for each storage group in the active SMS configuration.

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP,OPDISDLY,45
```

This updates the values for OPTICALREINITMODE and OPTICALDISPATCHERDELAY only for storage group GROUP22.

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP,OPREINIT,OAMSCR
```

This updates the values for OPTICALREINITMODE twice for storage group GROUP22. The first update takes place followed by the second. The final result is OAMSCR.

Using the UPDATE Command to Set OAMXCF Values

The F OAM,UPDATE command is used to allow users to update the OAMXCF timeout value settings without having to restart OAM. This command can also define settings of the OAMXCF timeout values if a CBROAM*xx* PARMLIB member was not used at OAM startup. This command provides dynamic processing that provides another method (other than using the CBROAM*xx* PARMLIB member) for customizing your object tape and optical support.

The OAM UPDATE command can be used to change the current settings for OAMXCF timeout parameters for the duration of the OAM session. If OAM is restarted, the settings from the CBROAM*xx* PARMLIB member override any OAMXCF values changed by an update command.

The following is the syntax for the F OAM,UPDATE,OAMXCF command:

```
►►──┬─MODIFY─┬──OAM,UPDATE,OAMXCF,parm,value,parm,value──────────────────────►◄
    └─F──────┘
```

The following are the descriptions of the keywords used in this command:

OAMXCF
Specifies the command parameter being updated. OAMXCF is used to update settings values in a Parallel Sysplex environment.

parm Specifies the timeout parameter setting that is being updated.

The following are valid timeout parameters:

OPTREADA
Indicates the number of seconds that an OAM originating an optical read request, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFOPTREADA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

OPTREADM
Indicates the number of seconds that an OAM originating an optical read request for a shelf-resident volume, which is shipped to another OAM within the OAMplex that owns the library where the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFOPTREADM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

OPTWRITA
Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains real (automated) optical libraries, which is shipped to another OAM within the OAMplex that owns the optical library defined to the object storage group for processing, should wait for completion of the write request. This parameter equates to the XCFOPTWRITEA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

OPTWRITM
Indicates the number of seconds that an OAM originating an optical write request targeted for an object storage group that contains pseudo libraries, which is shipped to another OAM within the OAMplex that owns the pseudo library defined to the object storage group for processing, should wait for completion of the write request. This parameter equates to the XCFOPTWRITEM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

TAPREADA
Indicates the number of seconds that an OAM originating a tape read request targeted for an automated tape library dataserver, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFTAPEREADA parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

TAPREADM
Indicates the number of seconds that an OAM originating a tape read request targeted for a manual tape library, which is shipped to another OAM within the OAMplex that owns the library in which the object resides for processing, should wait for completion of the read request. This parameter equates to the XCFTAPEREADM parameter of the OAMXCF statement in the CBROAM*xx* member of PARMLIB.

value Specifies the numeric value, in seconds, that is to be used for the specified timeout parameter.

Note: Valid input values for the OAMXCF timeout parameters are numeric values in the range of 0–999999. A time-out value of zero will cause the originating OAM to wait until a request is completed and a response is returned from the target OAM without ever timing out the request.

Updating Fields in the DB2 Volume Table and the Tape Volume Table

The F OAM,UPDATE,VOLUME command is used to allow users to update specific fields in a volume record related to an optical or tape volume used for object storage by OAM. This command may be used in place of using SPUFI to make the changes, and then stopping and starting the OAM address space in order for the changes to be reflected.

Note: The command requires that the volume requested for the update request must reside in a library that is either currently managed and controlled by the OAM on which the command was entered or not controlled and managed by any OAM at this time.

The following MVS command syntax updates valid fields in the DB2 Volume Table or the Tape Volume Table:

```
►►──┬─MODIFY─┬──OAM,UPDATE,─┬─VOLUME─┬─,volser,field,value,field,value──────────►◄
    └─F──────┘              └─VOL────┘
```

The following are the descriptions of the keywords used in this command:

UPDATE Specifies that an update is requested to a volume record related to an optical or tape volume used for object storage.

VOLUME Indicates whether this update command is for an optical or tape volume.

volser Indicates the specific volume serial number targeted for the update.

field Specifies the specific field in the DB2 Volume Table or the Tape Volume Table that is targeted for the update.

value Specifies the value to be assigned to the field as a result of the update.

Table 40 shows the valid fields, field values, and value descriptions for updating optical volumes:

Table 40. Field Values for Optical Volumes

FIELD	VALID VALUES	DEFINITION OF VALUE
EMPTY	Y N	Indicates the volume is empty. Indicates the volume is not empty.
LOSTFLAG	OFF	The volume lost flag will be reset. This is the only valid value for the LOSTFLAG operand.
EXPDATE	yyyymmdd	A valid date that specifies the scheduled expiration date for the volume. **Note:** If all objects on this volume are not yet expired or deleted, OSMC automatically recalculates this date when the volume is selected for expiration, unless this volume belongs to the Object Backup storage group. This value may be recalculated if an object is stored to this volume prior to the expiration of the volume.
FULL	Y N	Indicates that this volume is full. Indicates that this volume is not full, or that it is not writable. (See 104 for a discussion of the TAPEPERCENTFULL parameter.)
READABLE	Y N	Indicates that this volume is readable. Indicates that this volume is not readable.
WRITABLE	Y N	Indicates that this volume can be written to. Indicates that this volume cannot be written to.
WRITPROT	Y N	Indicates that this volume is write protected. Indicates that this volume is not write protected.

Table 41 shows the valid fields, field values, and value descriptions for updating object tape volumes:

Table 41. Field Values for Object Tape Volumes

FIELD	VALID VALUES	DEFINITION OF VALUE
LOSTFLAG	OFF	The volume lost flag will be reset. This is the only valid value for the LOSTFLAG operand.

Table 41. *Field Values for Object Tape Volumes (continued)*

FIELD	VALID VALUES	DEFINITION OF VALUE
EXPDATE	yyyymmdd	A valid date that specifies the scheduled expiration date for the volume. **Note:** If all objects on this volume are not yet expired or deleted, OSMC automatically recalculates this date when the volume is selected for expiration, unless this volume belongs to the Object Backup storage group. This value may be recalculated if an object is stored to this volume prior to the expiration of the volume.
FULL	Y	Indicates that this volume is full.
	N	Indicates that this volume is not full, or that it is not writable. (See page 104 for a discussion of the TAPEPERCENTFULL parameter.) **Note:** If you mark a volume full and you want it to remain marked full, then you need to increase the PFULL value to 100%. Otherwise, OAM will mark the volume not full during initialization.
PFULL	0–100	Decimal value in the range of 0 to 100 that specifies the current percentage full for this object tape volume. This value may be recalculated by OAM initialization or after a write request to this volume.
READABLE	Y N	Indicates that this volume is readable. Indicates that this volume is not readable.
WRITABLE	Y N	Indicates that this volume is writable. Indicates that this volume is not writable.

Auditing a Volume

OAM provides an AUDIT command that enables the system operator to audit a library resident tape or optical volume.

The syntax of the command for the AUDIT function is:

```
►►──┬─MODIFY─┬──OAM,──AUDIT──,──┬─VOLUME, volser──────────────┬──────────────►◄
    └─F──────┘                  ├─VOLLIST, volser1,volser2...─┤
                                └─LIBRARY,library-name────────┘
```

The following are the descriptions of the keywords used in this command:

VOLUME | VOLLIST | LIBRARY
: Specifies the scope (single volume, volume list, or all volumes in a library) of the audit to be performed.

volser
: Specifies the single tape or optical volume to be audited when the scope is VOLUME.

volser1, volser2...
: Specifies up to 15 tape or optical volumes to be audited when the scope is VOLLIST.

library-name
: Specifies the name of the tape or optical library to be audited when the scope is LIBRARY.

Remapping an Optical Library

OAM provides a REMAP command that enables the system operator to remap an optical library.

The syntax of the command for the REMAP function is:

```
►►──┬─MODIFY─┬──OAM,──REMAP,──library-name──────────────────────────────►◄
    └─F──────┘
```

library-name
Specifies the name of the optical library that is to be remapped.

Stopping OAM Functions

Use the OAM STOP command to stop:
- OAM
- OSMC
- OSMC processing for a particular storage group
- Move volume processing for a particular volume
- Access backup processing

The following command syntax stops OAM:

```
►►──┬─MODIFY─┬──OAM,──┬─STOP─┬──,──┬─OAM─────────────────────────┬──────────────►◄
    └─F──────┘        └─P────┘     ├─OSMC────────────────────────┤
                                   ├─STORGRP,storage-group-name──┤
                                   ├─MOVEVOL,volser──────────────┤
                                   └─AB,REASON───────────────────┘
```

Attention: To stop OAM, DB2 must be active. If you plan to stop DB2, stop OAM first.

OAM
Stops OAM and stops OSMC if it is running. OSMC does not complete work in process before stopping.

OSMC
OSMC completes all work currently in process before stopping. OSMC continues processing DISPLAY commands.

STORGRP,*storage-group-name*
OSMC completes all work currently in process for this Object storage group before stopping.

MOVEVOL,*volser*
OSMC completes movement of objects that are currently in the process of being moved before stopping.

AB,REASON
Automatic access to backup copies of objects on removable media is discontinued for the specified reason.

Stopping OAM

To stop OAM, enter one of the following commands:

```
F OAM,STOP,OAM
   or
STOP OAM
```

The system displays messages indicating OAM termination status. If you stop OAM and the OSMC cycle is running, you receive the following messages:

```
CBR1000I OAM STOP command execution scheduled.

CBR0098I OAM termination starting.

CBR9011I OAM requested OSMC to terminate.

CBR9012I OSMC completed termination.

CBR0074I OAM XCF member xcf-member-name successfully left OAM XCF group
         xcf-group-name

CBR0099I OAM termination completed.
```

Note: If OSMC is running and OSMC processing is still completing, you also receive other OSMC messages. OSMC does not complete work in process before stopping.

Stopping OSMC

To stop OSMC, enter the following command:

```
F OAM,STOP,OSMC
```

The system displays messages indicating OSMC completion status. When OSMC is running, you also see OSMC messages as processing completes, in addition to the following messages:

```
CBR9047I Operator requested OSMC to stop processing.

CBR1000I OAM STOP command execution scheduled.

CBR9010I OSMC has stopped.
```

Note: OSMC will wait for outstanding requests sent to LCS to process completely before stopping. This is not a CANCEL type of stop.

Stopping the Move Volume Utility

To stop the Move Volume utility for a volume, enter the following command:

```
F OAM,STOP,MOVEVOL,volser
```

where *volser* is the volume serial of the source volume from which objects are being moved.

The system issues the following messages:

```
CBR9856I Move Volume Utility stopping for volume volser.

CBR9858I Move Volume Utility status for volume volser. Total: total
         Attempted: attempted, Successful: successful,
         Unsuccessful:  unsuccessful.

CBR9859I Move Volume Utility ending for volume  volser.
```

The Move Volume utility completes any work it is currently processing, but does not move more objects than those already completed or those the utility is currently processing.

Stopping Access Backup

To stop access backup processing, enter the following command:

```
F OAM,STOP,AB,reason
```

The following are valid values and descriptions for the *reason* keywords:

UNREAD
Automatic access to backup processing is stopped for object retrieves from unreadable media. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is marked not readable, the retrieve request fails.

OFFLINE
Automatic access to backup processing is stopped for object retrieves from resident volumes (optical or tape media) in libraries that are not online. When a retrieve for an object is attempted and the volume on which the object resides is in a library that is currently offline or pending offline, the retrieve request fails.

NOTOPER
Automatic access to backup processing is stopped for object retrieves from resident volumes (optical or tape media) in libraries that are not operation. When a retrieve for an object is attempted and the media on which the object resides is in a library that is currently marked nonoperational, the retrieve request fails.

ALL
This is the default. Automatic access to backup processing is stopped for all object retrieves. When a retrieve for an object is attempted and the volume (optical or tape media) on which the object resides is not available for any of the above reasons the retrieve request fails.

The system displays the following messages indicating access backup processing is stopped:

```
CBR1000I OAM STOP command execution scheduled.

CBR1091I OAM Access Backup processing stopped for reason.
```

Stopping OTIS

To stop OTIS, enter one of the following commands:

Note: Remember that if any OAM applications are processing the OTIS address space must be active.

```
F OTIS,STOP
    or
STOP OTIS
```

When the OTIS address space has ended, the system issues the following message:

```
CBR8511I OTIS subsystem has terminated.
```

Appendix A. Sample Optical Hardware Configurations

This appendix provides information on the following topics:

These topics discuss the 9246/9247 Optical Storage Subsystem and the IBM 3995 Optical Library Dataserver. This appendix provides the following information for each device:

- The major hardware components
- An overview of the physical connections between the various components
- Information important to the system software

MVS/ESA 9246/9247 Optical Storage Subsystem

The 9246/9247 Optical Storage Subsystem consists of a FileNet Optical Disk Storage and Retrieval Library Unit (OSAR 64 Model 100) with two Laser Magnetic Storage International (LMSI) LaserDrive 1250E Optical Disk Drives. Additional RPQs (customized hardware configurations) that are available include the following configurations:

- The MVS/ESA Direct Attachment RPQ, which attaches the library unit to MVS/ESA.
- The number of optical disk drives in the IBM 9246 may be increased from two to four drives.
- The LaserDrive 1200E may function as a stand-alone drive.

Figure 34 on page 324 shows a sample 9246/9247 Optical Storage Subsystem hardware configuration.

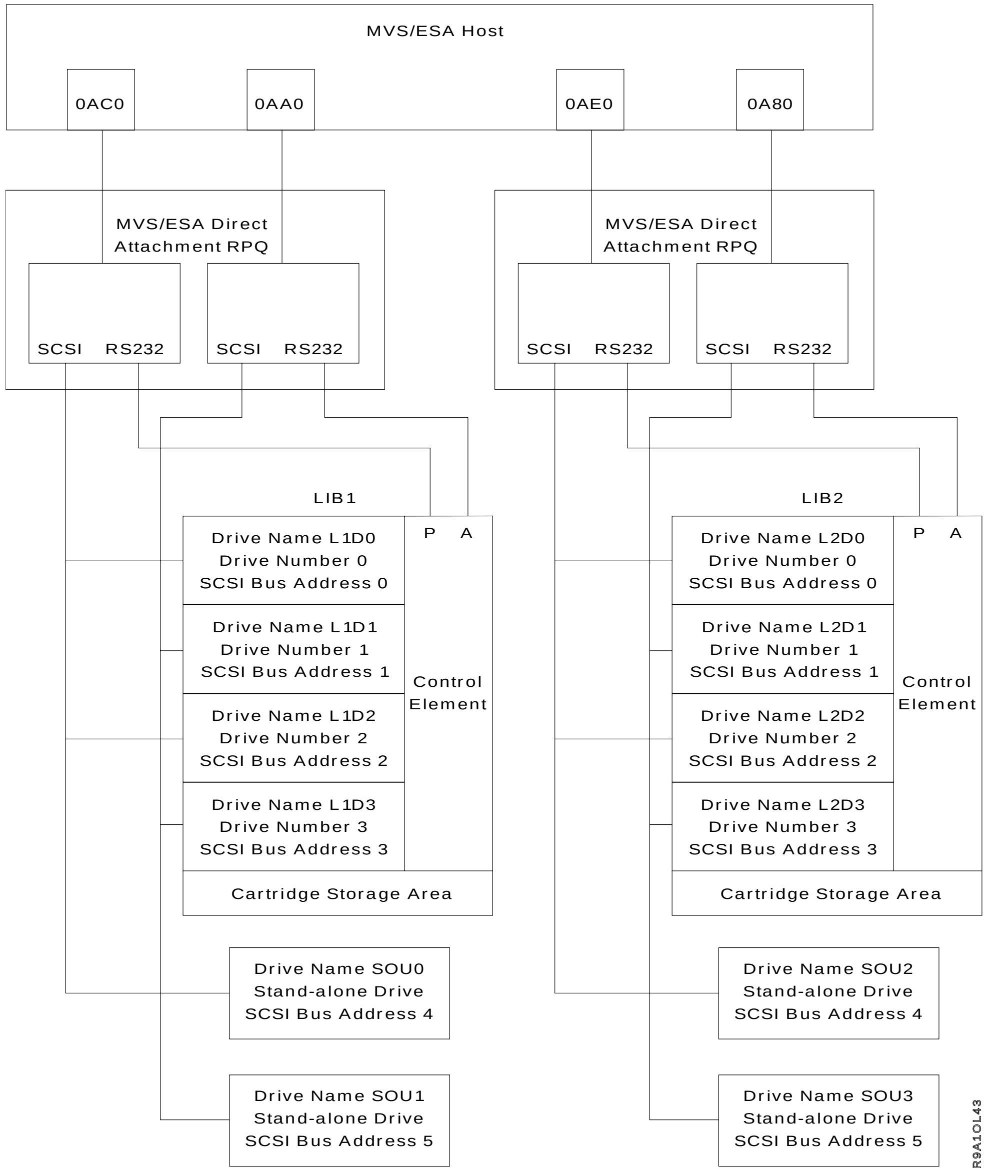

Figure 34. Sample Hardware Configuration for the 9246/9247 Optical Storage Subsystem

Notes to Figure 34:

1. SCSI is a small computer system interface.
2. RS232 is an Electronics Industries Association (EIA) standard for serial interfaces between computers and communication equipment.
3. P=primary and A=alternate control ports.

MVS/ESA Direct Attachment RPQ

The IBM MVS/ESA Direct Attachment RPQ is a modified IBM 8232 Model 2 Channel Station. The IBM 8232 Model 2 Channel Station contains two IBM 7532 Industrial PC/ATs. Each 7532 contains a personal computer channel adapter (PCCA) card. The PCCA card operates similarly to the IBM 3088 Multisystem Channel Communication Unit (MCCU) consisting of 32 channel-to-channel adapters (CTCAs). Each 7532 attaches, via the PCCA card, to a System/370™* block multiplexer channel. The 7532s can be attached to the same block multiplexer channel or to different block multiplexer channels. Each PCCA card operates in one of three modes: high speed DC-interlock, 1.0 megabyte/second data streaming, or

3.0 megabytes/second data streaming. Each 7532 must be defined to the MVS/ESA operating system as 32 channel-to-channel adapters. The MVS device numbers for the 32 channel-to-channel adapters must be consecutive, and the first device number must be a multiple of 32.

Determining CTC Device Numbers

Before installing OAM, determine if sufficient channel-to-channel devices (CTCs) are defined and available. Each MVS/ESA Direct Attachment RPQ contains two PCCA cards. Each PCCA card requires that 32 channel-to-channel adapters be defined; therefore, each MVS/ESA Direct Attachment RPQ requires that 64 channel-to-channel adapters be defined to the MVS operating system and the channel subsystem.

For the sample optical storage subsystem hardware configuration illustrated in Figure 34 on page 324, 128 channel-to-channel adapters must be defined to the MVS operating system and the channel subsystem.

Defining CTC Device Numbers

If not enough CTC adapters are defined in your I/O configuration, you must define the MVS/ESA Direct Attachment RPQ to the MVS operating system and the channel subsystem.

The MVS configuration program defines the I/O configuration to the MVS operating system. The input/output configuration program (IOCP) defines the I/O configuration to the channel subsystem. The following statements (see Figure 35) can be used as sample input to the MVS configuration program and the IOCP.

```
CUCTC1   CNTLUNIT CUNUMBR=0A80,PATH=(01),SHARED=N,UNIT=3088,               X
               UNITADD=((80,32)),PROTCOL=S
CTCA80   IODEVICE CUNUMBR=0A80,ADDRESS=(0A80,32),UNIT=3088,                X
               FEATURE=370
CUCTC2   CNTLUNIT CUNUMBR=0AA0,PATH=(01),SHARED=N,UNIT=3088,               X
               UNITADD=((A0,32)),PROTCOL=S
CTCAA0   IODEVICE CUNUMBR=0AA0,ADDRESS=(0AA0,32),UNIT=3088,                X
               FEATURE=370
CUCTC3   CNTLUNIT CUNUMBR=0AC0,PATH=(01),SHARED=N,UNIT=3088,               X
               UNITADD=((C0,32)),PROTCOL=S
CTCAC0   IODEVICE CUNUMBR=0AC0,ADDRESS=(0AC0,32),UNIT=3088,                X
               FEATURE=370
CUCTC4   CNTLUNIT CUNUMBR=0AE0,PATH=(01),SHARED=N,UNIT=3088,               X
               UNITADD=((E0,32)),PROTCOL=S
CTCAE0   IODEVICE CUNUMBR=0AE0,ADDRESS=(0AE0,32),UNIT=3088,                X
               FEATURE=370
```

Figure 35. Defining CTC Device Numbers to the MVS Operating System and Channel Subsystem

These CNTLUNIT and IODEVICE statements are consistent with Figure 34 on page 324, where one MVS/ESA Direct Attachment RPQ attaches to the MVS host at device numbers 0AC0 and 0AA0, and the other attaches to the MVS host at device numbers 0AE0 and 0A80.

Before using these sample input statements, verify that the device numbers specified on the IODEVICE statements are consistent with your existing I/O configuration. Because no two devices can have the same device number, your existing I/O configuration should not have any devices defined with a device number in the ranges listed. If your existing I/O configuration has any devices defined with device numbers in the ranges listed, either change your I/O configuration or change the sample IODEVICE and CNTLUNIT statements.

Attaching to a 3044 Fiber-Optic Channel Extender Link

The MVS/ESA Direct Attachment RPQ can be attached to an IBM 3044 fiber-optic channel extender link. The MVS/ESA Direct Attachment RPQ can be attached to the downstream unit (D01) of a 3044 Model 1 or the downstream unit (D02) of a 3044 Model 2.

If the MVS/ESA Direct Attachment RPQ is attached to the downstream unit (D01) of a 3044 Model 1, the channel speed of the 7532 should be set to 1 megabyte/second data streaming mode when the functional microcode is installed. For information of the installation of the functional microcode in the 7532, refer to *RPQ Optical Storage Subsystem Product User's Manual*.

If the MVS/ESA Direct Attachment RPQ is attached to the downstream unit (D02) of a 3044 Model 2, then the channel speed of the 7532 should be set to 3.0 megabytes/second data streaming mode when the functional microcode is installed.

Optical Libraries

The sample optical storage subsystem hardware configuration illustrated in Figure 34 on page 324 contains two optical libraries, named LIB1 and LIB2. The libraries are defined using the ISMF library configuration application, and the names of the libraries can be from 1 to 8 characters. This library name is used in messages to the operator when an event occurs related to a specific optical library.

Each optical library contains a cartridge storage area, cartridge transport mechanism, control element, and up to four optical disk drives.

Communication with the control element, which controls the cartridge transport mechanism, occurs on one of two serial ports using an RS232 interface. One serial port is designated as primary and the other is designated as alternate. If the host system cannot communicate with the control element of the optical library using the device connected to the primary port, the host tries to use the device connected to the alternate port.

The device attached at device number 0AC0 is physically connected to the primary port of library LIB1. The device attached at device number 0AA0 is physically connected to the alternate port of library LIB1. The device attached at device number 0AE0 is physically connected to the primary port of library LIB2. The device attached at device number 0A80 is physically connected to the alternate port of library LIB2.

Optical Disk Drives

The sample optical storage subsystem hardware configuration illustrated in Figure 34 on page 324 contains 12 optical disk drives. Four drives are in each of the two libraries and four are stand-alone models. The drives in a library are Laser Magnetic Storage International LaserDrive 1250 optical disk drives. The stand-alone drives are Laser Magnetic Storage International LaserDrive 1200 optical disk drives.

Use the ISMF drive configuration application to define the optical disk drives. The names of the drives can be from one to eight characters. This drive name is used in messages to the operator when an event occurs related to a specific optical disk drive.

Each optical disk drive within the optical library is assigned a library drive number. The library drive number is from 0 to 3 and indicates the physical position of the drive within the optical library. The library drive number labels are inside the optical library, next to each optical disk drive.

Communication with the optical disk drives is through the small computer system interface (SCSI) bus. Each device on an SCSI bus is assigned an address and a logical unit number.

The device address of an optical disk drive comprises:
- The device number of the optical disk controller to which the optical disk drive is attached
- The SCSI bus address of the optical disk drive
- The library drive number of the optical disk drive

OAM uses the SCSI bus address of each optical disk drive to access the optical disk drive during I/O operations. OAM uses the library drive number for optical cartridge movement operation.

Table 42 lists pertinent information about the 12 optical disk drives.

Table 42. Optical Disk Drive Information

Drive Name	Library Name	Device Number	SCSI Bus Address	Library Drive Number
L1D0	LIB1	0AC0	0	0
L1D1	LIB1	0AA0	1	1
L1D2	LIB1	0AC0	2	2
L1D3	LIB1	0AA0	3	3
SOU0		0AC0	4	
SOU1		0AA0	5	
L2D0	LIB2	0AE0	0	0
L2D1	LIB2	0A80	1	1
L2D2	LIB2	0AE0	2	2
L2D3	LIB2	0A80	3	3
SOU2		0AE0	4	
SOU3		0A80	5	

ISMF Session for a Sample 9246 Library and 9247 Drive Configuration

The following series of panels illustrates an ISMF session defining a sample 9246 optical library and 9247 optical drive configuration. From the ISMF Primary Option menu for storage administrators (as shown in Figure 36 on page 328), select option **10** to start the library management dialog.

```
  Panel  Help
------------------------------------------------------------------------------
                           ISMF PRIMARY OPTION MENU - DFSMS 1.5
Enter Selection or Command ===>

Select one of the following options and press ENTER:

 0  ISMF Profile              - Specify ISMF User Profile
 1  Data Set                  - Perform Functions Against Data Sets
 2  Volume                    - Perform Functions Against Volumes
 3  Management Class          - Specify Data Set Backup and Migration Criteria
 4  Data Class                - Specify Data Set Allocation Parameters
 5  Storage Class             - Specify Data Set Performance and Availability
 6  Storage Group             - Specify Volume Names and Free Space Thresholds
 7  Automatic Class Selection - Specify ACS Routines and Test Criteria
 8  Control Data Set          - Specify System Names and Default Criteria
 9  Aggregate Group           - Specify Data Set Recovery Parameters
 10 Library Management        - Specify Library and Drive Configurations
 11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
 C  Data Collection           - Process Data Collection Function
 L  List                      - Perform Functions Against Saved ISMF Lists
 R  Removable Media Manager   - Perform Functions Against Removable Media
 X  Exit                      - Terminate ISMF
 Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 36. ISMF Primary Option Menu

Defining 9246 Optical Libraries

When you select Library Management, the Library Management Selection menu is displayed, as shown in Figure 37 on page 329. At this point, you can select either an optical library configuration, an optical drive configuration, or tape library configuration. Since this manual deals with OAM's relationship with optical libraries, only the optical options are discussed. For more information on OAM's role with tape libraries, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Because a library must be defined before any devices associated with that library can be defined, you must first define libraries and then define drives. Selecting option 1 causes the Optical Library Application Selection panel to be displayed.

```
  Panel  Help
 ------------------------------------------------------------------------------
                      LIBRARY MANAGEMENT SELECTION MENU
 Enter Selection or Command ===>_

  1  Optical Library         - Optical Library Configuration
  2  Optical Drive           - Optical Drive Configuration
  3  Tape Library            - Tape Library Configuration















Use HELP Command for Help; Use END Command to Exit.
```

Figure 37. Library Management Selection Menu

In this sample configuration, as shown in Figure 38, two libraries are defined: one real library named LIB1, and one pseudo library named STDALONE. The name of the source control data set (SCDS) that contains the library and drive definitions is SCDS.PRIMARY. To define the first library called LIB1, enter **LIB1** into the LIBRARY NAME field, enter **9246** in the LIBRARY DEVICE TYPE field, and enter option **3**.

```
  Panel  Utilities  Help
 ------------------------------------------------------------------------------
                    OPTICAL LIBRARY APPLICATION SELECTION
 Command ===>

 To Perform Library Operations, Specify:
   CDS Name . . . . . . . 'SCDS.PRIMARY'
                                (1 to 44 Character Data Set Name or 'ACTIVE')
   Library Name . . . . . LIB1  (For Optical Library List , fully or
                                    Partially Specified or * for All)
   Library Device Type . . 9246  (For Optical Library List, fully or
                                    Partially Specified or * for All)
   Library Type . . . . . REAL  (REAL, PSEUDO, or * for ALL)

Select one of the following Options:

    3 1. List    - Generate a list of Libraries
      2. Display - Display a Library
      3. Define  - Define a Library
      4. Alter   - Alter a Library
If List Option is Chosen,
   Enter "/" to select option    _  Respecify View Criteria
                                 _  Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 38. Optical Library Application Selection Panel

Note: After the first library definition, ISMF primes CDS NAME, LIBRARY NAME, and LIBRARY DEVICE TYPE with the last used reference values on the

Optical Library Application Selection panel. To define a library, you must specify the name of an SCDS in CDS NAME, provide a name in the LIBRARY NAME field, and a device type in the LIBRARY DEVICE TYPE field, and specify the library type in the LIBRARY TYPE field. (The default CDS NAME is the single-quoted word 'ACTIVE', which represents the currently active configuration.)

Choose option **3** from the Optical Library Application Selection panel to display the 9246 Library Define panel with all the input fields blanked out. You can then enter a sample LIB1 definition, as shown in Figure 39. When the panel is complete, press the END key.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              9246 LIBRARY DEFINE
COMMAND ===>_

SCDS Name . : SCDS.PRIMARY
Library Name: LIB1
Library Type: REAL

To Define Library, Specify:
  Description ==>
              ==>

  Online Status . . . . . . . Y       (Y or N)
  Current Path  . . . . . . . PRIMARY (PRIMARY or ALTERNATE)
  Primary Device Number . . . 0AC0    (Valid CTC device number)
  Primary Port Address  . . . 1       (1 or 2)
  Alternate Device Number . . 0AA0    (Valid CTC device number)
  Alternate Port Address  . . 1       (1 or 2)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 39. 9246 Library Define Panel

Note: You can leave the 9246 Library Define panel at any time without saving optical library attributes by issuing the **CANCEL** command.

The following are output fields specified in the Optical Library Application Selection panel:

SCDS NAME
The name of the source control data set this library will be defined in.

LIBRARY NAME
The name of the library being defined.

LIBRARY TYPE
Specifies the type of library you are addressing. This type can either be REAL or PSEUDO. A REAL optical library is a physical library containing two or more drives. A PSEUDO optical library is the set of shelf-resident optical disk volumes associated with optical stand-alone, or operator-accessible disk drives. If you specify PSEUDO in the library type, also specify ONLINE=N for offline.

In order to process read and write requests from optical shelf-resident volumes, you need to define compatible optical stand-alone or operator-accessible drives to the pseudo optical library when defining it to the optical configuration database. If you do not define compatible optical stand-alone or operator-accessible drives to the pseudo optical library, OAM cannot process read and write requests for optical shelf-resident volumes.

If the optical configuration database does not define a pseudo library for this type of 9246 real library, then OAM assigns a default pseudo optical library for the optical shelf-resident volumes after ejecting the volumes from the real library. The default pseudo optical library is STDALONE for the 9246. For more information on pseudo libraries, refer to "Pseudo Optical Library Concept" on page 34.

Specify the following information in the optical configuration database and make an entry in the current source control data set. See Figure 38 on page 329 for an example.

DESCRIPTION
A 120-byte field that allows you to enter a description of the library definition for use by the installation. There are no restrictions on its content.

ONLINE STATUS
A 1-character field showing whether the library is online or offline when this SCDS is activated: **Y** (YES) for online; **N** (NO) for offline. The library status is set to this value each time this SCDS is activated.

CURRENT PATH
PRIMARY or ALTERNATE. Specify PRIMARY, which means that the system first tries to use the primary path to access the library.

PRIMARY DEVICE NUMBER
See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

PRIMARY PORT ADDRESS
See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

ALTERNATE DEVICE NUMBER
See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

ALTERNATE PORT ADDRESS
See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

After you complete the definition of LIB1, the Optical Library Application Selection menu (Figure 38 on page 329) is displayed again, and you can define a pseudo library. Enter the name of the pseudo library, **STDALONE**, into the library name field and press ENTER. The 9246 Library Define panel (Figure 40 on page 332) is again displayed, and you enter **PSEUDO** in the library type field. For pseudo libraries, enter only the library type field, because none of the other fields apply.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              PSEUDO LIBRARY DEFINE
Command ===>_

SCDS Name  . : SCDS.PRIMARY
Library Name : STDALONE
Library Type : PSEUDO

To Define Library, Specify:
  Description ==>
              ==>









Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 40. Pseudo Library Define Panel for Defining a 9246 Pseudo Library—STDALONE

Altering 9246 Optical Libraries

To alter a 9246 Library, display the Optical Library Application Selection panel Figure 38 on page 329. Specify the CDS NAME, LIBRARY NAME, LIBRARY DEVICE TYPE, and the LIBRARY TYPE for the library you wish to alter, and then select option **4** (Alter). The 9246 Library Alter Panel (Figure 41) is displayed.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              9246 LIBRARY ALTER
COMMAND ===>_

SCDS Name . : SCDS.PRIMARY
Library Name: LIB1
Library Type: REAL

To ALTER Library, Specify:
  Description ==>
              ==>

  Online Status . . . . . . . Y       (Y or N)
  Current Path  . . . . . . . PRIMARY (PRIMARY or ALTERNATE)
  Primary Device Number . . . 0AC0    (Valid CTC device number)
  Primary Port Address  . . . 1       (1 or 2)
  Alternate Device Number . . 0AA0    (Valid CTC device number)
  Alternate Port Address  . . 1       (1 or 2)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 41. 9246 Library Alter Panel

Any of the following fields can be altered to change the library definition in the SCDS for the 9246 library.

DESCRIPTION
: A 120-byte field that allows you to change a description of the library definition for use by the installation. There are no restrictions on its content.

ONLINE STATUS
: Allows you to change whether the library should be online or offline to the currently logged on system when this SCDS is activated. **Y** (YES) for online. **N** (NO) for offline. The library status is set to this value each time this SCDS is activated.

CURRENT PATH
: To change which path the system should try first, change the specification for this field. This can be either PRIMARY or ALTERNATE. Specifying PRIMARY means that the system first tries to use the primary path to access the library.

PRIMARY DEVICE NUMBER
: See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

PRIMARY PORT ADDRESS
: See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

ALTERNATE DEVICE NUMBER
: See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

ALTERNATE PORT ADDRESS
: See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of these numbers.

Defining 9247 Optical Drives

After you complete the previous library definitions, you have defined one real library, LIB1, along with the pseudo library, STDALONE. Now, you can define drives for each of these libraries. For this sample configuration, one drive will be defined for each library. This is all done within the same SCDS named SCDS.PRIMARY. From the Library Management Selection menu, (Figure 37 on page 329), select option **2**, (Optical Drive Configuration), to display the Optical Drive Application Selection menu shown in Figure 42 on page 334. For each of the drives to be defined, enter the drive name into the DRIVE NAME field, **9247** into the DRIVE DEVICE TYPE field, and choose option **3** to continue with the definition.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                    OPTICAL DRIVE APPLICATION SELECTION
Command ===>_

To Perform Drive Operations, Specify:
  CDS Name . . . . . . . 'SCDS.PRIMARY'
                          (1 to 44 Character Data Set Name or 'ACTIVE')
  Drive Name . . . . . . L1D0  (For Optical Drive List, fully or
                               Partially Specified or * for All)
  Drive Device Type  . . 9247  (For Optical Drive List, fully or
                               Partially Specified or * for All)

Select one of the following options:
  3  1. List    - Generate a list of Drives
     2. Display - Display a Drive
     3. Define  - Define a Drive
     4. Alter   - Alter a Drive

If List Option is Chosen,
   Enter "/" to select option   _ Respecify View Criteria
                                _ Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 42. Optical Drive Application Selection Panel

Note: After the first drive definition, ISMF primes CDS NAME, DRIVE NAME, and DRIVE DEVICE TYPE with the last used reference values on the Optical Drive Application Selection panel. To define a drive, you must specify the name of an SCDS in CDS NAME, provide a name in the DRIVE NAME field, and a device type in the DRIVE DEVICE TYPE field. (The default CDS NAME is the single-quoted word ‘ACTIVE’, which represents the currently active configuration.) The ‘ACTIVE’ SCDS cannot be modified.

Choose option **3** to display the 9247 Drive Define panel with all the input fields blanked out. You can then enter a sample drive definition, as shown in Figure 43 on page 335 for LIB1 or Figure 44 on page 336 for STDALONE. When the panel is complete, press the END key.

Note: You can leave the Drive Define panel at any time without saving optical disk drive attributes by issuing the **CANCEL** command.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             9247 DRIVE DEFINE
Command ===>_

SCDS Name  :SCDS.PRIMARY
Drive Name :L1D0

To Define Drive, Specify:

  Description ==>
              ==>

  Library Name  . . LIB1             (1 to 8 characters)
  Drive Type  . . . LIBRARY          (LIBRARY or STDALONE)
  Drive Number  . . 0                (0 to 3 or blank for STDALONE)
  Online Status . . Y                (Y or N)
  Device Number . . 0AC0             (Valid CTC device number)
  SCSI Address  . . 0                (0 to 7)



Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 43. 9247 Drive Define Panel for L1D0

SCDS NAME and DRIVE NAME are output fields that contain the SCDS and drive names you specified in the Optical Drive Application Selection panel.

Specify the following information in the optical configuration database and make an entry in the current SCDS:

DESCRIPTION
A 120-byte field that allows you to enter a description of the drive definition for use by the installation. There are no restrictions on its content.

LIBRARY NAME
Specify the 1- to 8-character name of the library to which this drive is assigned.

DRIVE TYPE
Specify the drive type by entering **LIBRARY** or **STDALONE**.

DRIVE NUMBER
Specify the drive number by selecting 0 to 3 for a drive within an optical library, or blank for a stand-alone drive.

ONLINE STATUS
A 1-character field showing whether the drive is online or offline when this SCDS is activated. **Y** (YES) for online. **N** (NO) for offline. The drive status is set to this value each time this SCDS is activated.

DEVICE NUMBER
See "Determining CTC Device Numbers" on page 325 and "Defining CTC Device Numbers" on page 325 for an explanation of this number.

SCSI ADDRESS
Specify the assigned SCSI bus address (0 to 7); see "Optical Disk Drives" on page 326 for more information.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            9247 DRIVE DEFINE
Command ===>_

SCDS Name  :SCDS.PRIMARY
Drive Name :SOU0

To Define Drive, Specify:

  Description ==>
              ==>

  Library Name  . . STDALONE          (1 to 8 characters)
  Drive Type  . . . STDALONE          (LIBRARY or STDALONE)
  Drive Number  . .                   (0 to 3 or blank for STDALONE)
  Online Status . . Y                 (Y or N)
  Device Number . . 0AC0              (Valid CTC address)
  SCSI Address  . . 4                 (0 to 7)



Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 44. 9247 Drive Define Panel for SOU0

The following names represent the libraries and their associated drive names:

Library	Drive Name
LIB1	L1D0
STDALONE	SOU0

MVS/ESA 3995 Optical Library Dataserver

There are various models of the IBM 3995 Optical Library Dataserver. In a nonsysplex environment where OAM is **not** running in an OAMplex with DB2 data sharing, each model can be connected to a single host processor operating in basic mode or to one logical partition of a processor complex operating in LPAR mode.

Attention: Multihost attachment is not supported and causes unpredictable results.

The 3995 models with two parallel channel adapters (PCA) are attachable to the host through parallel channels, while 3995 models with one or two ESCON® channel adapters (ECA) are attachable to the host through ESCON channels.

Up to two ESCON directors can be connected between the 3995 Optical Library Dataserver and the host, but only one ESCON director may use dynamic link connection. The 3995 ESCON models may connect to the host in one of the following manners.

When using dynamic link, the host side port number of the dynamic link is required by the 3995 configuration program. Refer to *IBM 3995 Optical Library Dataserver: Maintenance Information* for more information regarding supplying the port numbers to the 3995 configuration file.

When using static link, or when not using an ESCON director, the host side port number is not required by the 3995 configuration program.

In an OAMplex with DB2 data sharing, it is recommended that you use an ESCON product environment and the System Automation for z/OS for connectivity as opposed to using parallel channels.

In a Parallel Sysplex environment it is possible to establish a multisystem connection (logical not physical) that is controlled so that there is only one physical library-host connection at a time. Multiple instances of OAM within an OAMplex can be connected to a single library-host connection. The logical connection to these instances of OAM can be changed by specifying which library on a specific OAM system should be connected to the host system. This logical connection allows data to be accessed from and shared between various libraries associated with multiple instances of OAM within an OAMplex environment.

Configurations for the 3995-1xx Models

Figure 45 on page 338 shows a sample hardware configuration for the IBM 3995 Optical Library Dataserver with two parallel channel adapters.

Note: Although Figure 45 on page 338 shows a configuration with two parallel channel adapters, most 3995-133 and 3995-113 Optical Library Dataservers are ESCON attached.

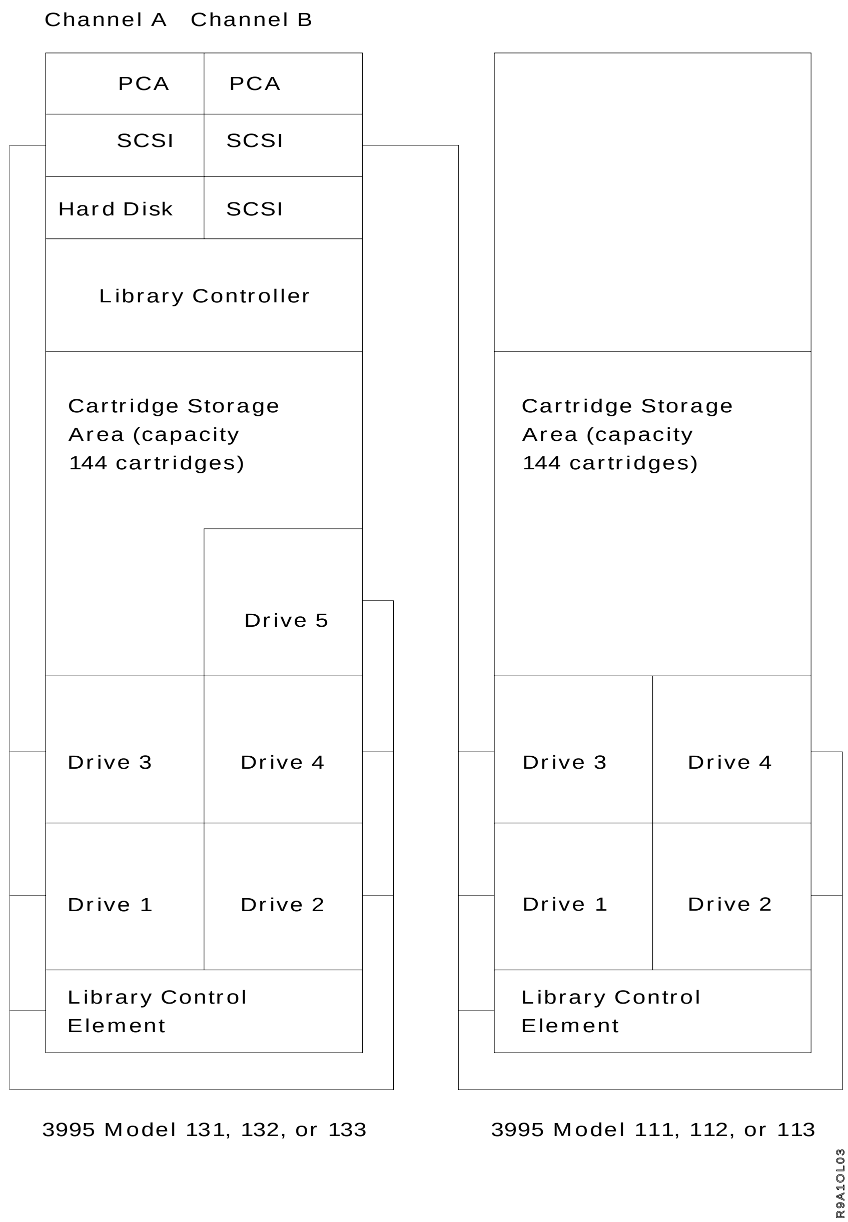

Figure 45. Sample Hardware Configuration—IBM 3995 Models 131, 132, 133, 111, 112, 111

Each channel adapter can be configured to run in one of the following modes, as shown in Table 43 via the IBM 3995 Optical Dataserver's RAS package.

Table 43. PCA Card Speed Setting to Match CPU Channel Speeds.

CPU rated channel speed	**Optical Library Controller PCA speed setting**
High speed DC interlock	1.5 MB/second
2.0 MB/second data streaming	1.9 MB/second
3.0 MB/second data streaming	2.7 MB/second
3.5 MB/second data streaming	3.4 MB/second
4.5 MB/second data streaming	4.5 MB/second
ESCON channel speed	17.5 MB/second

Configurations for the 3995-Cxx Models

Figure 46 on page 340, Figure 47 on page 341, Figure 48 on page 342, and Figure 49 on page 343 show sample hardware configurations for the 3995 C-Series Models. The C32, C34, C36, and C38 are attached to a single host processor through two ESCON channels or two parallel channels. While neither attachment provides multiple host support, ESCON attachment does support remote connection of the 3995 up to 3 kilometers to the first director or host. For a configuration example for the 3995-11x and 3995-13x models see “Configurations for the 3995-1xx Models” on page 337.

Note: If you are running SMS in a shared sysplex environment with other DFSMS levels (prior to DFSMS 1.4.0) that have 3995-Cxx models defined, toleration PTFs are required. Refer to the *z/OS Program Directory* for specific details.

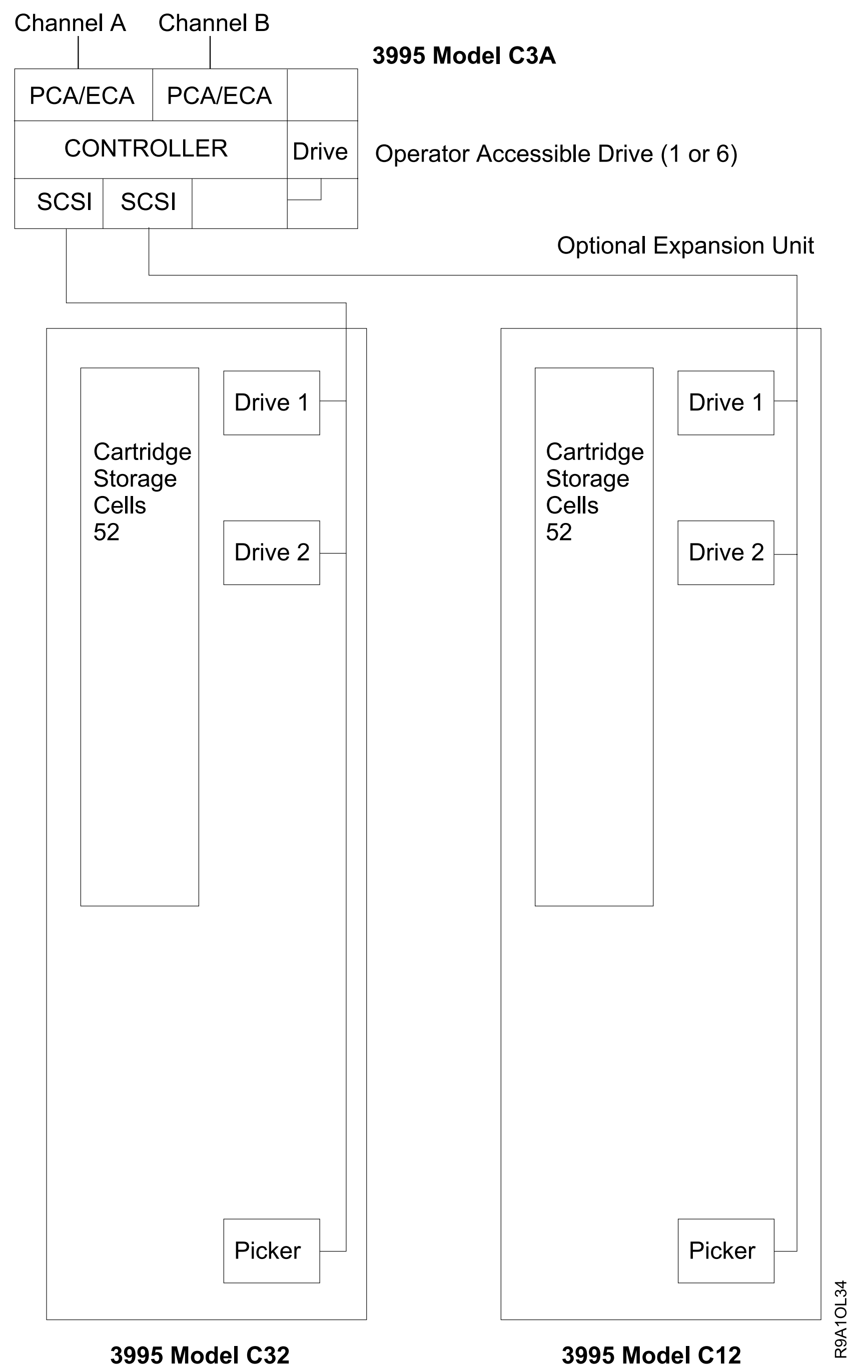

Figure 46. Sample Hardware Configuration—IBM 3995 Models C3A, C32, and C12. This configuration requires twelve addresses.

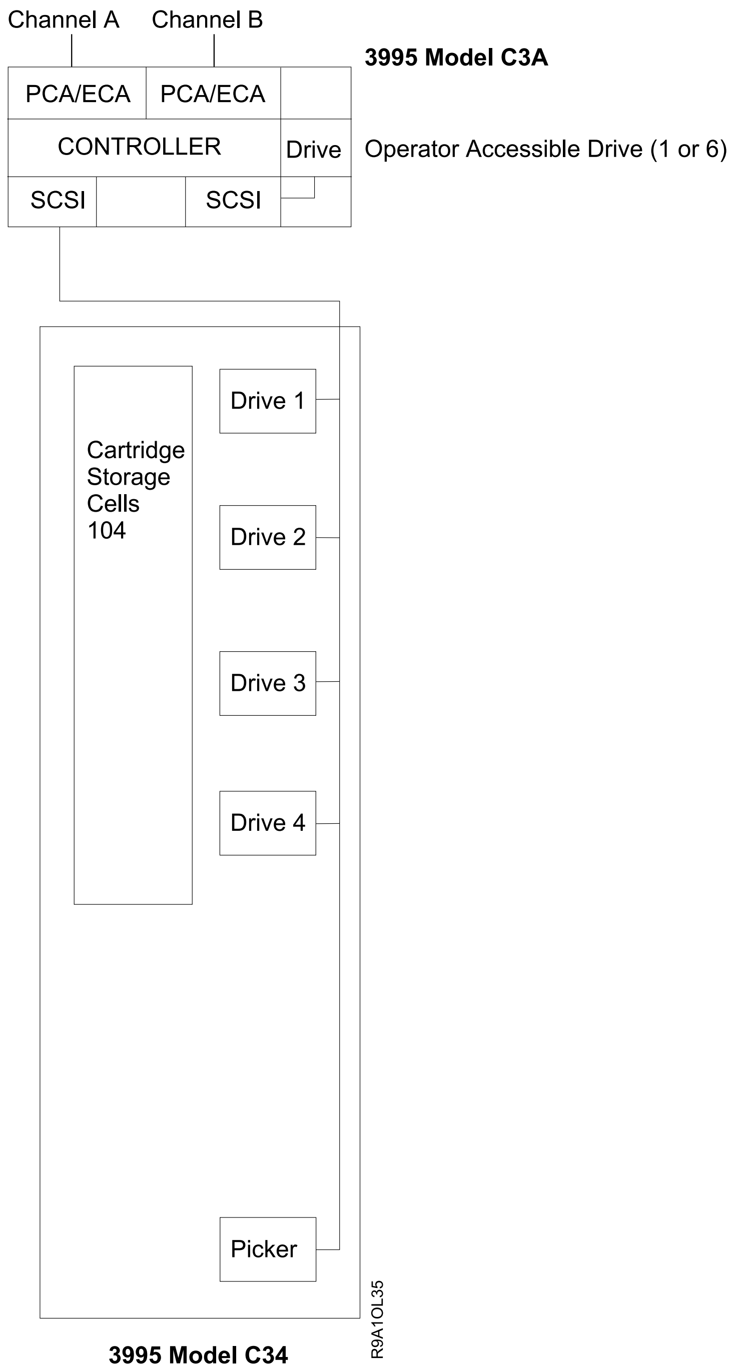

Figure 47. Sample Hardware Configuration—IBM 3995 Models C3A and C34. This configuration requires ten addresses.

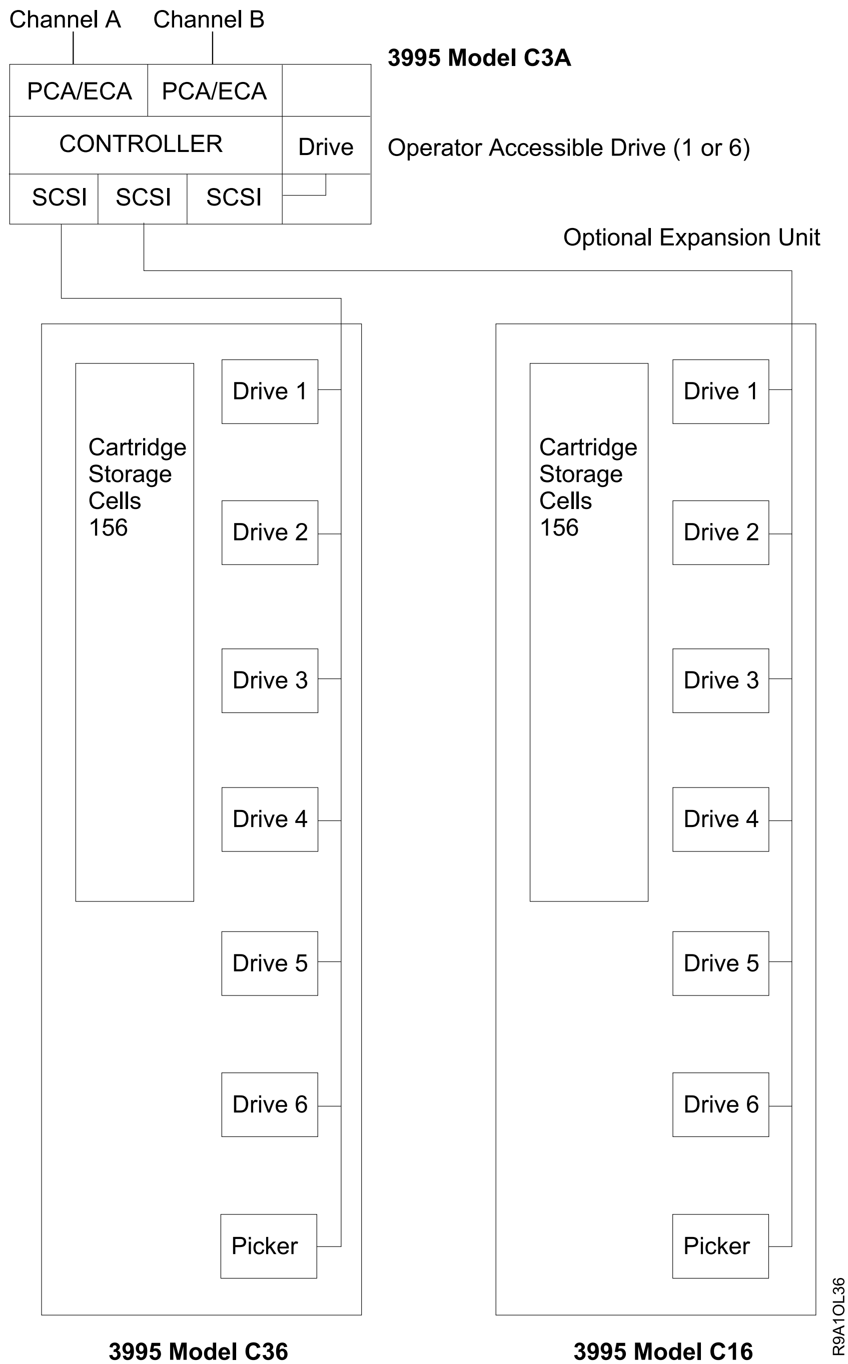

Figure 48. Sample Hardware Configuration—IBM 3995 Model C3A, C36, and C16. This configuration requires twenty addresses.

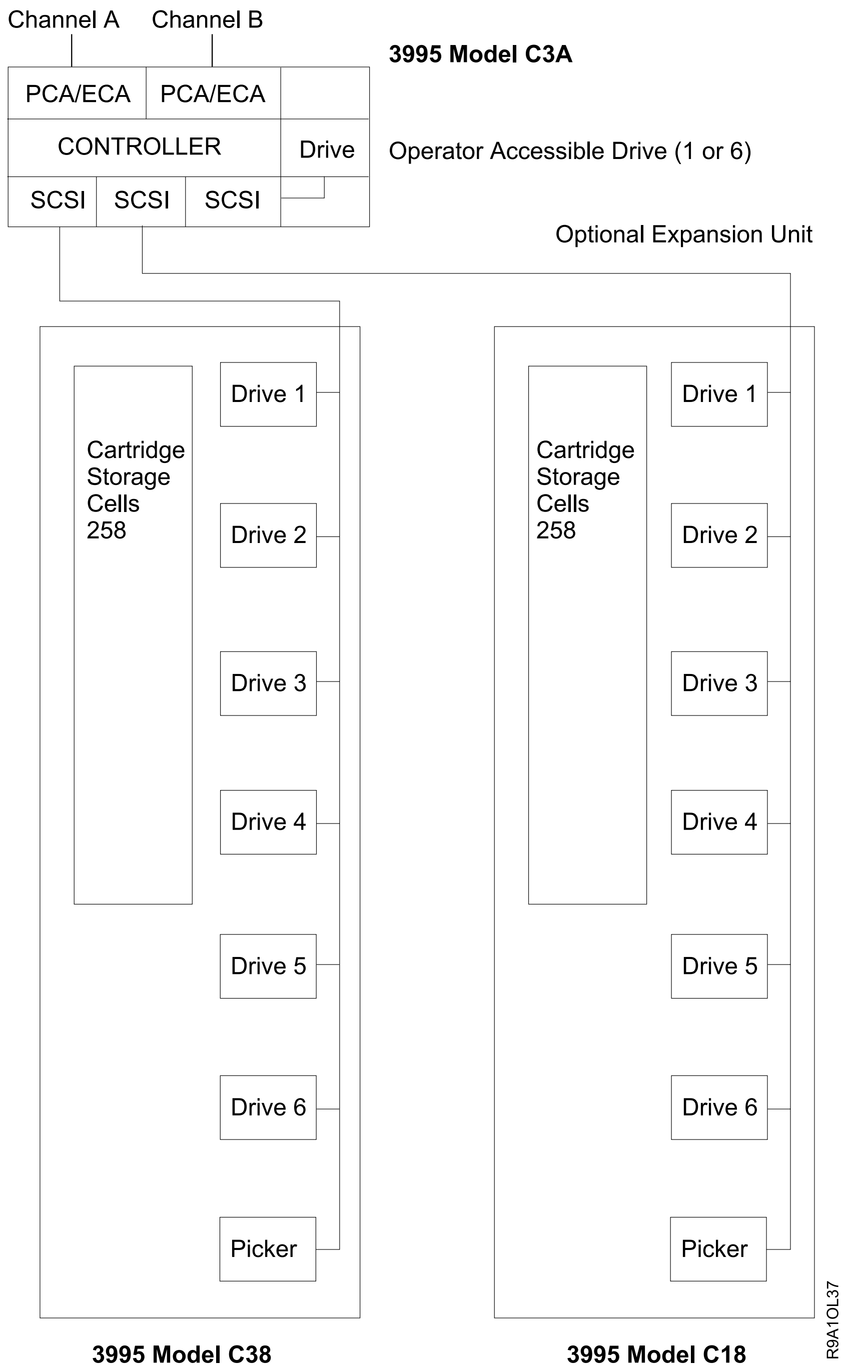

Figure 49. Sample Hardware Configuration—IBM 3995 Model C3A, C38 and C18. This configuration requires twenty addresses.

Defining 3995 Device Numbers

OAM uses specific MVS/ESA base device numbers for communicating the following with the optical disk library and the optical disk drives within the library:

- Sending library related commands to the 3995 library
- Receiving unsolicited attention interrupts from the input output station
- Performing input and or output to specific optical disk drives

Each IBM 3995 library configuration is required to have device numbers defined to the MVS operating system and channel subsystem. A library configuration can consist of a controller, a single library, or a single library with an expansion unit. *The number of device numbers depends on the hardware configuration.*

Attention: Multihost attachment from two separate processors or two separate partitions of the same processor is not supported in a nonOAMplex environment. Only one active host-library connection is supported in the multihost logical connection environment of an OAMplex.

The base device number for the IBM 3995-13x or 3995-C3A controlling library models must be a multiple of 16, meaning the low order digit of the base device number must be zero, for example, 0940.

The base device number for the IBM 3995-11x expansion unit models must be equal to the base device number of the controlling library (the IBM 3995-13x models) + 8, for example, 0948. The base device number for the expansion units (3995-11x models) is automatically calculated and does not need to be specified when defining these libraries.

The base device number for the IBM 3995-Cxx models must match the base device number assigned to the device when it was installed. For further information on defining the 3995-Cxx libraries, see Figure 57 on page 358.

With MVS/ESA SP release 5 or later, you *must* use the HCD panels to define the new 3995 device numbers. You cannot use the macro instructions for MVS Configuration Program (MVSCP).

Defining 3995 device numbers with HCD provides the following capabilities:
- Dynamic I/O capability
- Specially designed 3995 Error Recovery Processing
- 3995 optical service information messages
- Ability to have MVS Unit Control Blocks for 3995 devices reside above the 16MB line in 31-bit addressable storage
- Assigning the device type of 3995 for all 3995 device numbers

Using 3995-SDA Definition in HCD

3995-SDA stands for a 3995 with self-description architecture. Self-description architecture allows the 3995 hardware to provide information concerning the optical library and optical drives to the OAM. This allows OAM to depend on information provided by the hardware without having to perform validity checking on its own. Although the 3995-13x models are not self-descriptive devices, the preferred method for defining all 3995 libraries is to use the 3995-SDA definition method.

To define the device numbers for a channel attached 3995 library, perform the following steps:
1. Use the Add Control Unit panel to define 3995-SDA type control units. Specify the base device number of each 3995 controller (3995-13x or 3995-C3A model) as a control unit. The base device number for 3995 library controller must be a multiple of 16, meaning the low order digit of this device number must end in zero, for example, 0940 or 0AC0. This is the control unit number that must be used when attaching the channel paths to the 3995. The unit addresses for the 3995 control unit must begin with 00 when attached to ESCON CHPIDs.
2. Use the Add Device panel to define the 3995. You must supply the starting device number; the low order digit of this device number must end in zero, for example 0940 or 0AC0. The device type must be a 3995-SDA. You can reserve

device numbers for future drive expansions of the library. This allows you to best utilize channel address resources while allowing for the planning of future upgrades of the drives. The number of devices must be greater than zero and less than or equal to 256. The default is 16. The control unit number is the same as the base device number. Using the default number of devices as an example, devices 0AC0–0ACF are generated.

Note: The 3995 does not belong to any esoteric device group and is not reserved through device allocation.

Using 3995 Definition in HCD

This section describes defining device numbers for 3995-1xx models.

Note: The 3995-SDA definition method is the preferred method for defining *any* 3995 library. For information concerning this definition method, see “Using 3995-SDA Definition in HCD” on page 344.

To define the device numbers for IBM 3995-1xx models, perform the following steps:

1. Use the Add Control Unit panel to define 3995 type control units. Specify the base device number of each 3995-13x model as a control unit. The base device number for the 3995-13x library must be a multiple of 16, meaning the low order digit of this device number must end in zero, for example, 0940 or 0AC0. This is the control unit number that must be used when attaching the channel paths to the 3995 Optical Dataserver. The unit addresses for the 3995 control unit must begin with 00 when attached to ESCON CHPIDs.
2. Use the Add Device panel to define the 3995-13x model. You must supply the base device number; the low order digit of this device number must end in zero, for example 0940 or 0AC0. The number of devices must be 1 and the device type must be a 3995. The control unit number is the same as the base device number. Using 0AC0 as an example, devices 0AC0–0AC7 are generated.
3. Use the Add Device Panel to define the 3995-11x model. You must supply the base device number; the low order digit of this device number must be 8 (the base device number for the controlling library, IBM 3995-13x models + 8), for example, 0948 or 0AC8. The number of devices must be 1 and the device type must be 3995. The control unit number is the same as the base number of the 3995-13x model to which this 3995-11x model is attached. Using 0AC8 as an example, devices 0AC8–0ACF are generated, with 0AC0 as the control unit device number.

   Note: The 3995 does not belong to any esoteric device group and is not reserved through device allocation.

Migrating the Configuration from IOCDS into IODF

The OAM optical device statements in the IOCP/MVSCP input data set must be upgraded for a migration to HCD/IODF for the following reasons:

- In contrast to IOCP, HCD validates device and control unit types, and associated parameters such as the protocols supported by a given control unit type and the connection of a device to a control unit.
- HCD introduced the specification of the SDA model for the 3995.
- HCD supports the true 3995 device type that had to be defined as a “look-like” device for MVSCP (3995 was defined as a “look-like” CTC, CTCS, or 3088 in MVSCP).

There are really two options that can be used to assist you in migrating to the HCD/IODF structure. The first method, and the preferred option, is to redefine your optical devices using HCD control unit and device definition panels. An alternate method is to change your MVSCP/IOCP statements and migrate them to IODF. Both methods are discussed in this section.

The *preferred* method of migrating your configuration from IOCDS to IODF is to perform the following steps:

1. Delete the CNTLUNIT and IODEVICE statements for the OAM 3995 optical devices from the old IOCP/MVSCP input data set.
2. Use the HCD migration panels or run a batch job to create a migrated IODF work file using the updated IOCP/MVSCP input from the previous step. This will create an IODF work file that does not include OAM 3995 optical devices.
3. Add the OAM 3995 optical devices to the IODF work file generated from the previous step by using the appropriate HCD panels. Use the HCD Control Unit Panel to add the OAM 3995 optical library controllers. Use the HCD I/O Device Panel to add the OAM 3995 optical devices.

An alternate method of migrating from MVSCP/IOCP to IODF is to change the device statements within the IOCP input data set. The HCD panels or a batch job can then be used to perform the migration.

Note: It is recommended that the *preferred* method be used instead of this alternate method. The changes made to the device and control unit statements in the alternate method are device specific and therefore may cause errors if not modified correctly.

The following examples are provided to assist you in changing your MVSCP/IOCP statements for 3995 optical devices for migration purposes only. These examples can be used when migrating an old MVSCP/IOCP input data set into an IODF file using the HCD migration panel or batch job. These statements should not be used as an input example for MVSCP or IOCP programs.

Figure 50 shows examples of changing the device and control unit statements for 3995-Cxx optical devices:

```
* ========================================================== *
* A. 3995 Cxx Series                                         *
**************************************************************
**************************************************************
*                                                            *
* A1. ESCON 3995-C3x/C1x optical devices                     *
*     using ESCON (serial attached to Host)                  *
*                                                            *
*           (such as:  3995-C32/C12, 3995-C34/C14,           *
*                      3995-C36/C16, 3995-C38/C18)           *
```

Figure 50. Changing the Device and Control Unit Statements for 3995-Cxx Optical Devices (Part 1 of 3)

```
***************************************************************
*                                                             *
*  >>> ESCON 3995-SDA CHPID    statement              <<<    *
*        --->Note:                                            *
*              SWITCH=ss (ss: ESCON Director Number)          *
*              TYPE=CNC                                       *
*                                                             *
*              Also define the PARTITION                      *
*              Reconfigurable (REC) if running in LPAR mode  *
*                                                             *
***************************************************************
 CHPID PATH=((7B)),SWITCH=0A,TYPE=CNC,PARTITION=(PC6,REC)
 CHPID PATH=((7F)),SWITCH=0B,TYPE=CNC,PARTITION=(PC6,REC)
***************************************************************
*  >>> ESCON 3995-SDA CNTLUNIT statement              <<<    *
*        --->Note:                                            *
*              UNITADD=((00,nnn))  (nnn: # of devices         *
*                                   1 < nnn =< 256 )          *
*              UNIT=3995-SDA                                  *
*              LINK=(l1,l2)        (l1: ESCON Link Address)  *
*                                  (l2: ESCON Link Address)  *
***************************************************************
 CNTLUNIT CUNUMBR=0005,PATH=(7B,7F),UNITADD=((00,032)),        +
       UNIT=3995-SDA,LINK=(C4,C5)
***************************************************************
*  >>> ESCON 3995-SDA I/O device (IODEVICE) statement  <<<   *
*        --->Note:                                            *
*              ADDRESS=((xxx0,nnn))  (nnn: # of devices       *
*                                     1 < nnn =< 256 )        *
*              UNITADD=00                                     *
*              TIMEOUT=N                                      *
*              UNIT=3995                                      *
*              MODEL=SDA                                      *
*              DYNAMIC=YES                                    *
*              LOCANY=YES                                     *
***************************************************************
 IODEVICE ADDRESS=(19D0,032),UNITADD=00,CUNUMBR=(0005),        +
       UNIT=3995,MODEL=SDA,DYNAMIC=YES,LOCANY=YES
***************************************************************
***************************************************************
*                                                             *
* A2. OEMI 3995-C3x/C1x optical devices                       *
*     using OEMI (parallel attached to Host)                  *
***************************************************************
*                                                             *
*  >>> OEMI  3995-SDA CHPID    statement              <<<    *
*        --->Note:                                            *
*              TYPE=BL                                        *
*                                                             *
***************************************************************
 CHPID PATH=((9D)),TYPE=BL
 CHPID PATH=((9E)),TYPE=BL
```

Figure 50. Changing the Device and Control Unit Statements for 3995-Cxx Optical Devices (Part 2 of 3)

```
*************************************************************
*  >>> OEMI 3995-SDA control unit (CNTLUNIT) statement <<<   *
*       --->Note:                                           *
*             UNITADD=((00,nnn))  (nnn: # of devices        *
*                                    1 < nnn =< 256 )       *
*             SHARED=N                                      *
*             PROTOCL=S4                                    *
*             UNIT=3995-SDA                                 *
*************************************************************
 CNTLUNIT CUNUMBR=0004,PATH=(9D,9E),UNITADD=((C0,016)),        +
       SHARED=N,PROTOCL=S4,UNIT=3995-SDA
*************************************************************
*  >>> OEMI 3995-SDA IO device (IODEVICE) statement    <<<   *
*       --->Note:                                           *
*             ADDRESS=((xxx0,nnn))  (nnn: # of devices      *
*                                      1 < nnn =< 256 )     *
*             TIMEOUT=N                                     *
*             UNIT=3995                                     *
*             MODEL=SDA                                     *
*             DYNAMIC=YES                                   *
*             LOCANY=YES                                    *
*************************************************************
 IODEVICE ADDRESS=(19C0,016),CUNUMBR=(0004),                   +
       TIMEOUT=N,UNIT=3995,MODEL=SDA,DYNAMIC=YES,LOCANY=YES
* ========================================================= *
```

Figure 50. Changing the Device and Control Unit Statements for 3995-Cxx Optical Devices (Part 3 of 3)

Figure 51 on page 349 shows examples of changing the device and control unit statements for 3995-1xx optical devices:

```
* ======================================================== *
*                                                          *
* B. 3995 1xx Series                                       *
*    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!    *
*    ***   IMPORTANT  IMPORTANT  IMPORTANT           ***    *
*    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!    *
*                                                          *
*          The changes required for 3995 1xx Series are the *
*          same as those for 3995 Cxx Series if your site   *
*          (1) is at DFSMS 1.4.0 or above level, or         *
*          (2) is at DFSMS 1.3.0 and has installed          *
*              the OAM SPE for 3995 Cxx Series support      *
*                                                          *
*          Follow the following changes for 3995 1xx        *
*          Series ONLY IF your site                         *
*                                                          *
*          (1) is at DFSMS 1.2.0 or below level, or         *
*          (2) is at DFSMS 1.3.0 and has not installed      *
*              the OAM SPE for 3995 Cxx Series support      *
*                                                          *
*    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!    *
*    ***   IMPORTANT  IMPORTANT  IMPORTANT           ***    *
*    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!    *
*                                                          *
************************************************************
************************************************************
*                                                          *
* B1. ESCON 3995-13x/11x optical devices                   *
*     using ESCON (serial attached to Host)                *
*                                                          *
*     (such as:  3995-133/113, 3995-132/112,               *
*                3995-131/111.)                            *
*                                                          *
************************************************************
*                                                          *
*  >>> ESCON 3995 channel path(CHPID) statement        <<< *
*        --->Note:                                         *
*              SWITCH=ss  (ss: ESCON Director Number)      *
*              TYPE=CNC                                    *
*                                                          *
*              Also must define the PARTITION as           *
*              reconfigurable (REC) if running in LPAR mode *
************************************************************
 CHPID    PATH=((A6)),TYPE=CNC,PART=(PC6,REC),SWITCH=07
 CHPID    PATH=((A7)),TYPE=CNC,PART=(PC6,REC),SWITCH=08
************************************************************
*                                                          *
*  >>> ESCON 3995 control unit(CNTLUNIT) statement     <<< *
*        --->Note:                                         *
*              UNITADD=((00,016))                          *
*              UNIT=3995                                   *
*              LINK=(l1,l2)          (ESCON Link Addresses) *
*                                                          *
************************************************************
 CNTLUNIT CUNUMBR=0164,PATH=(A6,A7),UNITADD=((00,016)),       +
       UNIT=3995,LINK=(ED,EF)
```

Figure 51. Changing the Device and Control Unit Statements for 3995-1xx Optical Devices (Part 1 of 3)

```
**************************************************************
*  >>> ESCON 3995-13x (A BOX) IODEVICE  statement:      <<<  *
*       (When the HCD migration completed successfully,      *
*        the following example statement will generate       *
*        8 device numbers for 3995 from 0E40 to 0E47)        *
*        --->Note:                                           *
*              ADDRESS=(xxx0,001)                            *
*              UNITADD=00                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
**************************************************************
 IODEVICE ADDRESS=(0E40,001),UNITADD=00,CUNUMBR=(0164),        +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
*  >>> ESCON 3995-11x (B BOX) IODEVICE  statement:      <<<  *
*         (When the HCD migration completed successfully,    *
*          the following example statement will generate     *
*          8 device numbers for 3995 from 0E48 to 0E4f)      *
*          --->Note:                                         *
*              ADDRESS=(xxx8,001)                            *
*              UNITADD=08                                    *
*              TIMEOUT=N                                     *
*              UNIT=3995                                     *
*              DYNAMIC=YES                                   *
*              LOCANY=YES                                    *
*                                                            *
*              Also the controller unit number (CUNUMBR)     *
*              of A and B boxes must be the same.            *
**************************************************************
 IODEVICE ADDRESS=(0E48,001),UNITADD=08,CUNUMBR=(0164),        +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
**************************************************************
* B2. OEMI 3995-13x/11x optical devices                      *
*     using OEMI (parallel attached to Host):                *
*     (such as:  3995-133/113, 3995-132/112,                 *
*                3995-131/111.)                              *
**************************************************************
*                                                            *
**************************************************************
*  >>> OEMI 3995 channel path (CHPID) statement       <<<   *
**************************************************************
 CHPID PATH=((1A)),TYPE=BL
 CHPID PATH=((9C)),TYPE=BL
**************************************************************
*  >>> OEMI 3995     control unit (CNTLUNIT) statement <<<   *
*        --->Note:                                           *
*              UNITADD=((x0,016)),                           *
*              SHARED=N                                      *
*              PROTOCL=S4                                    *
*              UNIT=3995                                     *
```

Figure 51. Changing the Device and Control Unit Statements for 3995-1xx Optical Devices (Part 2 of 3)

```
**************************************************************
 CNTLUNIT CUNUMBR=00C5,PATH=(9C,1A),UNITADD=((50,016)),        +
       SHARED=N,PROTOCL=S4,UNIT=3995
**************************************************************
*  >>> OEMI 3995-13x (A BOX) IODEVICE  statement:       <<<  *
*                                                            *
*        (When the HCD migration completed successfully,     *
*         the following example statement will generate      *
*         8 device numbers for 3995 from 0950 to 0957)       *
*         --->Note:                                          *
*               ADDRESS=(xxx0,001)                           *
*               UNITADD=x0                                   *
*               TIMEOUT=N                                    *
*               UNIT=3995                                    *
*               DYNAMIC=YES                                  *
*               LOCANY=YES                                   *
**************************************************************
 IODEVICE ADDRESS=(0950,001),CUNUMBR=(00C5),                   +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
**************************************************************
*                                                            *
*  >>> OEMI 3995-11x (B BOX) IODEVICE  statement:       <<<  *
*                                                            *
*          (When the HCD migration completed successfully,   *
*           the following example statement will generate    *
*           8 device numbers for 3995 from 0958 to 095f)     *
*           --->Note:                                        *
*               ADDRESS=(xxx8,001)                           *
*               UNITADD=x8                                   *
*               TIMEOUT=N                                    *
*               UNIT=3995                                    *
*               DYNAMIC=YES                                  *
*               LOCANY=YES                                   *
*                                                            *
*               Also the controller unit number (CUNUMBR)    *
*               of A and B boxes must be the same.           *
**************************************************************
 IODEVICE ADDRESS=(0958,001),CUNUMBR=(00C5),                   +
       TIMEOUT=N,UNIT=3995,DYNAMIC=YES,LOCANY=YES
* ========================================================== *
```

Figure 51. Changing the Device and Control Unit Statements for 3995-1xx Optical Devices (Part 3 of 3)

Sample ISMF Session for an IBM 3995 Optical Library Dataserver

The following series of panels illustrates an ISMF session defining a sample 3995 library and drive configuration. These panels will illustrate how to define libraries and drives for 3995-1xx, 3995-Cxx, and pseudo libraries. From the ISMF Primary Option menu for storage administrators (as shown in Figure 52 on page 352), select option **10** to start the library management dialog.

```
  Panel  Help
------------------------------------------------------------------------------
                           ISMF PRIMARY OPTION MENU - DFSMS 1.5
Enter Selection or Command ===>

Select one of the following options and press ENTER:

 0  ISMF Profile              - Specify ISMF user profile
 1  Data Set                  - Perform Functions Against Data Sets
 2  Volume                    - Perform Functions Against Volumes
 3  Management Class          - Specify Data Set Backup and Migration Criteria
 4  Data Class                - Specify Data Set Allocation Parameters
 5  Storage Class             - Specify Data Set Performance and Availability
 6  Storage Group             - Specify Volume Names and Free Space Thresholds
 7  Automatic Class Selection - Specify ACS Routines and Test Criteria
 8  Control Data Set          - Specify System Names and Default Criteria
 9  Aggregate Group           - Specify Data Set Recovery Parameters
 10 Library Management        - Specify Library and Drive Configurations
 11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
 C  Data Collection           - Process Data Collection Function
 L  List                      - Perform Functions Against Saved ISMF Lists
 R  Removable Media Manager   - Perform Functions Against Removable Media
 X  Exit                      - Terminate ISMF

Use HELP Command for Help; Use END Command or X to Exit.
```

Figure 52. ISMF Primary Option Menu

When you select Library Management, the Library Management Selection menu is displayed, as shown in Figure 53. At this point, you select either an optical library configuration or an optical drive configuration. Because a library must be defined before any drives associated with that library can be defined, you must first define libraries and then define drives. Select option **1** to display the Optical Library Application Selection panel, as shown in Figure 54 on page 354.

```
  Panel  Help
------------------------------------------------------------------------------
                      LIBRARY MANAGEMENT SELECTION MENU
Enter Selection or Command ===>_

 1  Optical Library          - Optical Library Configuration
 2  Optical Drive            - Optical Drive Configuration
 3  Tape Library             - Tape Library Configuration

Use HELP Command for Help; Use END Command to Exit.
```

Figure 53. Library Management Selection Menu

The ISMF screens that follow provide examples for defining 3995-11x, 3995-13x, 3995-Cxx, and pseudo libraries. These examples define the pseudo libraries based on the media type and device type association; however, you can define pseudo libraries to best suit the requirements of your environment (based, for example, on location, or all backup objects with primary active objects, or data categories, and so forth). Where applicable, the ISMF screens are duplicated with examples for the different library definitions. The following libraries are defined in our sample configuration:

- **LIBRARY1**—3995-133
- **LIB1C**—3995-C3A
- **LIBRARY2**—3995-113
- **LIBRARYA**—3995-C38
- **LIBRARYB**—3995-C18
- **PSEUDO1**—operator-accessible drives.

The name of the source control data set (SCDS) that contains the library and drive definitions in these examples is SCDS.PRIMARY. The default CDS NAME is the single-quoted word ‘ACTIVE’, which represents the currently active configuration. Whenever you define a new optical library within an SCDS, the library definition added to the SCDS is identified by the system to which the storage administrator is logged on. You can use the Optical Library Configuration application to add an optical library to an SCDS. The Optical Library Configuration application must be running on the system on which the optical configuration database resides. The OAM optical configuration database cannot be shared among systems.

The OPTICAL LIBRARY APPLICATION SELECTION panel, Figure 54 on page 354, provides an example of selecting the define option to define a 3995-133 optical library (LIBRARY1). The following information must be provided on this screen:

- CDS Name—**‘SCDS.PRIMARY’**
- Library Name—**LIBRARY1**
- Library Device Number—**3995-133**
- Library Type—**REAL**
- Select Option **3** (Define) to continue the definition

```
  Panel  Utilities  Help
 ------------------------------------------------------------------------------
                     OPTICAL LIBRARY APPLICATION SELECTION
Command ===>_

To Perform Library Operations, Specify:
  CDS Name  . . . . . . . 'SCDS.PRIMARY'
                                 (1 to 44 character data set name or 'ACTIVE')
  Library Name  . . . . . LIBRARY1  (For Optical Library List, fully or
                                     Partially Specified or * for All)
  Library Device Type . . 3995-133  (For Optical Library List, fully or
                                     Partially Specified or * for All)
  Library Type . . . . .  REAL  (REAL, PSEUDO, or * for ALL)

Select one of the following Options:
  3  1. List    - Generate a list of Libraries
     2. Display - Display a Library
     3. Define  - Define a Library
     4. Alter   - Alter a Library
If List Option is Chosen,
   Enter "/" to select option    _ Respecify View Criteria
                                 _ Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 54. Optical Library Application Selection Panel

The following are output fields specified in the Optical Library Application Selection panel:

SCDS NAME
The name of the source control data set that this library will be defined in.

LIBRARY NAME
The name of the library being defined.

LIBRARY TYPE
The type of library you will be addressing. This can be either REAL or PSEUDO. A REAL optical library is a physical library containing optical disk drives and optical volumes that reside physically inside the library. A PSEUDO optical library is a set of shelf-resident optical disk volumes associated with stand-alone, or operator-accessible optical disk drives, or both.

In order to process read and write requests from optical shelf-resident volumes, you need to define compatible optical stand-alone or operator-accessible drives to the pseudo optical library when defining it to the optical configuration database. If you do not define any optical stand-alone or operator-accessible drives to the pseudo optical library, OAM cannot process read and write requests for optical shelf-resident volumes.

OAM stills supports the old concept of a pseudo optical libraries for each supported optical drive device type. If no pseudo optical library is defined in the active SMS configuration for a supported optical drive device type, OAM defines a default using the following names that can be used for assigning shelf-resident optical volumes:

- **PCTREUSE**—3995-131 device types
- **PCTWORM**—3995-132 device types
- **P3995133**—3995-133 device types
- **P3995SW3**—3995-SW3 device types
- **P3994SW4**—3995-SW4 device types

For more information on pseudo libraries, refer to “Pseudo Optical Library Concept” on page 34.

Note: After the first library definition, ISMF primes CDS NAME, LIBRARY NAME, LIBRARY DEVICE TYPE, and LIBRARY TYPE with the last used reference values on the Optical Library Application Selection panel.

Defining Real 3995 Libraries

Choose option **3** from the Optical Library Application Selection panel to display the 3995 Library Define panel with all the input fields blanked out. You can then enter a sample definition, as shown in Figure 55 and Figure 56 on page 357 (for LIBRARY1), and as shown in Figure 57 on page 358 and Figure 58 on page 359 (for LIB1C). When the panel is complete, press the END key.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To DEFINE Library, Specify:

  Description ==>
              ==>

  Model Number . . . . .  133
  Base Device Number  . . 0940      (Valid base device number)
  Controlling Library . .           (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUD01   (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 55. 3995 Library Define Panel for LIBRARY1 (Page 1 of 2)

The following are field descriptions for the 3995 Library Define panel (1 of 2). The information provided on this panel is stored in the optical configuration database and an entry is made in the current source control data set:

DESCRIPTION
A 120-byte field that allows you to enter a description of the library definition for use by the installation. There are no restrictions on its content.

BASE DEVICE NUMBER
Specify the base device number of a IBM 3995 controlling library model. This field is required for all *controlling* library models. The low order digit of this base device number must be 0 (for example, 1AC0).

The base device number for the IBM 3995-11x expansion unit models must be equal to the base device number of the controlling library (the IBM 3995-13x models) + 8, for example, 0948. The base device number for the expansion units (3995-11x models) is automatically calculated and does not need to be specified when defining these libraries.

The base device number for the IBM 3995-Cxx models must match the base device number assigned to the device when it was installed. See "Defining 3995 Device Numbers" on page 343 for more information on the device numbers.

CONTROLLING LIBRARY

Specify the name of the 3995 control unit (**3995-13x** or **3995-C3A**) to which the 3995 library expansion unit is connected. This field is required for all connected 3995 library expansion unit models. The following lists the controlling library and the associated expansion unit:

3995-131
3995-111

3995-132
3995-112

3995-133
3995-113

3995-C3A
3995-C32, 3995-C34, 3995-C36, 3995-C38 3995-C12, 3995-C16, 3995-C18

Note: The 3995-C12, 3995-C16, and 3995-C18 must be attached to its corresponding 3995-C32, -C36, and -C38 before being attached to the 3995-C3A controlling library. See Table 3 on page 25 for more details concerning these devices.

The system connectivity defined for the controlling library is inherited by the libraries attached to it.

DEFAULT MEDIA TYPE

Specifying a default media type limits the type of media that can be entered into the specified optical library dataserver. It is also used as a criteria for the output volume selection for a grouped write request when using a multifunction library that is referenced by the Object or the Object Backup storage group to which the grouped write request is being written. The valid values for the default media type are:

3995
Any 3995 5.25-inch single-, double-, quad-, or 8x-density, WORM, or rewritable optical disk media. This is the default.

3995REWR
3995 5.25-inch, single-, double-, quad-, or 8x-density rewritable optical disk media.

3995WORM
3995 5.25-inch, single-, double-, quad-, or 8x-density WORM optical disk media.

3995-1
Only 3995 5.25-inch, single-density, WORM or rewritable optical disk media.

3995-1RW
Only 3995 5.25-inch, single-density, rewritable optical disk media.

3995-1WO
Only 3995 5.25-inch, single-density, WORM optical disk media.

3995-2
Only 3995 5.25-inch, double-density, WORM or rewritable optical disk media.

3995-2RW
Only 3995 5.25-inch, double-density, rewritable optical disk media.

3995-2WO
Only 3995 5.25-inch, double-density, WORM optical disk media.

3995-4
Only 3995 5.25-inch, quad-density, rewritable or WORM optical disk media.

3995-4RW
Only 3995 5.25-inch, quad-density, rewritable optical disk media.

3995-4WO	Only 3995 5.25-inch, quad-density, WORM optical disk media.
3995-8	Only 3995 5.25-inch, 8x-density, rewritable or WORM optical disk media.
3995-8RW	Only 3995 5.25-inch, 8x-density, rewritable optical disk media.
3995-8WO	Only 3995 5.25-inch, 8x-density, WORM optical disk media.

Notes:

1. Double-density, quad-, and 8x-density, WORM includes CCW media.
2. CCW is continuous composite WORM media. WORM is write-once-read-many media.
3. The 3995-SW3 drives (used within the C3A, C1x, and C3x libraries) are capable of only reading single-density WORM or rewritable media. A 3995-SW3 cannot handle write requests to this media. It is capable of reading from and writing to all other 3995 optical media types. The 3995-SW4 drives used within these libraries are capable of only reading from single- or double-density WORM or rewritable media. However, these drives are capable of reading from and writing to quad- or 8x-density WORM or rewritable media. Keep this in mind when deciding the appropriate default media type for an optical library.

DEFAULT PSEUDO LIBRARY

The name of the pseudo library to which the volume will be assigned after it is ejected from this real library.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 56. 3995 Library Define Panel for LIBRARY1 (Page 2 of 2)

The following is a description of the information required for the 3995 Library Define panel (2 of 2). The information provided on this panel is stored in the optical configuration database and an entry is made in the current source control data set:

INITIAL ONLINE STATUS

Indicates the library connectivity to specified systems when this SCDS is activated. The library status is set to this value each time this SCDS is

activated. The library must be connected to at least one system. The library only can be online to one system at a time. The default is blank.
- **Y** (YES) for online.
- **N** (NO) for offline.
- **Blank** for not connected.

Figure 57 shows how to define a 3995-C3A controlling library (LIB1C). The following information must be provided:
- CDS Name—**'SCDS.PRIMARY'**
- Library Name—**LIB1C**
- Library Device Number—**3995-C3A**
- Select Option **3** (Define) to continue the definition

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIB1C
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C3A
  Base Device Number  . . 0900      (Valid base device number)
  Controlling Library . .           (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.           (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 57. 3995 Library Define Panel for LIB1C (Page 1 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIB1C
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 58. 3995 Library Define Panel for LIB1C (Page 2 of 2)

After you complete the library definitions (for LIBRARY1, LIB1C, or both) the Optical Library Application Selection menu (Figure 54 on page 354) is displayed again, and you can define your next library. Enter the name of the library into the library name field, the model number into the model number field, and select option **3**. The 3995 Library Define panel is again displayed. Figure 59 and Figure 60 on page 360, Figure 61 on page 360 and Figure 62 on page 361, and Figure 63 on page 361 and Figure 64 on page 362 provide sample definitions for LIBRARY2 (3995-113), LIBRARYA (3995-C38), and LIBRARYB (3995-C18).

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY2
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  113
  Base Device Number  . . 0948     (Valid device number)
  Controlling Library . . LIBRARY1 (Library Name)        (if expansion unit)
  Default Media Type  . . 3995WORM
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 59. 3995 Library Define Panel for LIBRARY2 (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY2
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 60. 3995 Library Define Panel for LIBRARY2 (Page 2 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYA
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C38
  Base Device Number  . . 0904     (Valid base device number)
  Controlling Library . . LIB1C    (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 61. 3995 Library Define Panel for LIBRARYA (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE             Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYA
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 62. 3995 Library Define Panel for LIBRARYA (Page 2 of 2)

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE             Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYB
Library Type :REAL

To Define Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  C18
  Base Device Number  . . 090C     (Valid base device number)
  Controlling Library . . LIB1C    (Library Name)       (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library. PSEUDO1  (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 63. 3995 Library Define Panel for LIBRARYB (Page 1 of 2)

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             3995 LIBRARY DEFINE              Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARYB
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 64. 3995 Library Define Panel for LIBRARYB (Page 2 of 2)

Defining Pseudo Libraries

After you complete the definition of the last real library, display the Optical Library Application Selection menu (Figure 54 on page 354) again, and you can then define your pseudo library. Enter the SCDS name, the Library Name, the Library Device Type (if it is an old default pseudo library), and the pseudo library type into the Library Type field, and then select option **3** (Define). Figure 65 is displayed.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                             PSEUDO LIBRARY DEFINE
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :PSEUDO1
Library Type :PSEUDO

To Define Library, Specify:
  Description ==> Department HRA functional testing data
              ==>








Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 65. Pseudo Library Define Panel for PSEUDO1

Hit ENTER to perform the pseudo library definition. This definition will be added to the SCDS.

Defining Additional Optical Libraries

You can copy existing optical library definitions and modify them to create new ones by using the COPY line operator from the Optical Library List panel. For more information on how to copy these existing optical library definitions, see “Copying Optical Library and Drive Definitions” on page 376.

Defining Optical Drives

After completing the previous library definitions, you have defined real libraries (LIBRARY1, LIBRARY2, LIBRARYA, LIBRARYB, and LIB1C) and a pseudo library (PSEUDO1). Now, you can define drives for each of these libraries. For this sample configuration, define one drive for each library (library LIB1C is not illustrated). Define two operator-accessible drives for the pseudo library. Define all of these libraries within the same SCDS named SCDS.PRIMARY.

The ISMF screens that follow provide examples for defining 3995-11x, 3995-13x, and 3995-SW3 (the drive used with all the 3995-Cxx models) drives. Where applicable, the ISMF screens are duplicated with examples for the different drive definitions.

Table 44 shows the names of the libraries and their associated drive names and drive device types as defined in the sample configuration:

Table 44. Optical Libraries and their Associated Drive Names and Drive Device Types

Library	Drive Name	Drive Device Type
LIBRARY1	LIB1D1	3995-133
LIBRARY2	LIB2D1	3995-113
LIBRARYA	LIBAD1	3995-SW3
LIBRARYB	LIBBD1	3995-SW3
PSEUDO1	OPA1	3995-133
PSEUDO1	OPDRV1	3995-SW3

From the Library Management Selection menu, (Figure 53 on page 352), select option **2** (Optical Drive Configuration), to display the Optical Drive Application Selection menu, as shown in Figure 66 on page 364. For each of the drives to be defined, enter the drive name into the drive name field, the drive device type into the drive device type field, and choose option **3** to continue with the definition.

Note: The optical drives inherit their online and system connectivity for the configuration from the library to which they belong, or in the case of operator-accessible drives, from their controlling library.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                     OPTICAL DRIVE APPLICATION SELECTION
Command ===>_

To Perform Drive operations, Specify:
  CDS Name  . . . . . . 'SCDS.PRIMARY'
                                (1 to 44 Character Data Set Name or 'ACTIVE')
  Drive Name  . . . . . LIB1D1     (For Optical Drive List, fully or
                                    Partially Specified or * for all)
  Drive Device Type . . 3995-133   (For Optical Drive List, fully or
                                    Partially Specified or * for all)

Select One of the following options:
  3  1. List    - Generate a list of Drives
     2. Display - Display a Drive
     3. Define  - Define a Drive
     4. Alter   - Alter a Drive

If List Option is Chosen,
  Enter "/" to select option    _ Respecify View Criteria
                                _ Respecify Sort Criteria
 Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 66. Optical Drive Application Selection Panel

Note: After the first drive definition, ISMF primes CDS NAME, DRIVE NAME, and DRIVE DEVICE TYPE with the last used reference values on the Optical Drive Application Selection panel. To define a drive, you must specify the name of an SCDS in CDS NAME, provide a name in the DRIVE NAME field, and a device type in the DRIVE DEVICE TYPE field. (The default CDS NAME is the single-quoted word ‘ACTIVE’, which represents the currently active configuration.) The ‘ACTIVE’ SCDS cannot be modified.

Choose option **3** to display the 3995 Drive Define panel with all input fields blanked out.

Note: You can leave the Drive Define panel at any time without saving optical disk drive attributes by issuing the CANCEL command.

Figure 67 on page 365 shows the drive being defined for LIBRARY1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIB1D1
Model Number  :133

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARY1             (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . Y            (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .               (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 67. 3995 Drive Define Panel for LIB1D1

SCDS NAME, DRIVE NAME, and MODEL NUMBER are output fields that contain the SCDS name, drive name, and model number of the drive device type you specified in the Optical Drive Application Selection panel.

Specify the following information in the optical configuration database and make an entry in the current SCDS:

DESCRIPTION
A 120-byte field that allows you to enter a description of the drive definition. There are no restrictions on its content.

LIBRARY NAME
A 1- to 8-character library name to which the drive is assigned. For operator-accessible drives, this field is the name of a pseudo optical library.

DRIVE NUMBER
A 3-character field representing the position the drive occupies in the library. This is a required field for 3995 models and should have the following values:

- 1 to 4 for library-resident drives; 5 for operator accessible 3995-1xx models
- 1 to 999 for all other 3995 models

OPERATOR ACCESSIBLE DRIVE
Specify if this drive is an operator-accessible drive that should be connected to a controlling library.

CONTROLLING LIBRARY NAME
Specify the name of the 3995-13x or -C3A model optical library to which the operator-accessible disk drive is connected.

Figure 68 on page 366 shows the drive definition for library LIBRARY2.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIB2D1
Model Number  :113

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARY2            (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N           (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .              (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 68. 3995 Drive Define Panel for LIB2D1

Figure 69 shows the drive definition for library LIBRARYA.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIBAD1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARYA           (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N          (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .             (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 69. 3995 Drive Define Panel for LIBAD1

Figure 70 on page 367 shows the drive definition for library LIBRARYB.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :LIBBD1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . LIBRARYB            (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . . N           (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . .              (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 70. 3995 Drive Define Panel for LIBBD1

Figure 71 shows the drive definition for the pseudo library, PSEUDO1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :OPA1
Model Number  :133

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . PSEUDO1            (1 to 8 characters)
  Drive Number  . . . . 5
  Operator Accessible Drive . . Y          (Y or N)

The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . . LIBRARY1     (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 71. 3995 Drive Define Panel for OPA1

Figure 72 on page 368 shows the drive definition for the pseudo library, PSEUDO1.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            3995 DRIVE DEFINE
Command ===>_

SCDS Name . . :SCDS.PRIMARY
Drive Name  . :OPDRV1
Model Number  :SW3

To Define Drive, Specify:

  Description ===>
              ===>

  Library Name  . . . . PSEUDO1            (1 to 8 characters)
  Drive Number  . . . . 1
  Operator Accessible Drive . .Y           (Y or N)

 The Following Field is for Operator Accessible Drive Type Only:
  Controlling Library Name . . LIB1C       (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 72. 3995 Drive Define Panel for OPDRV1

Defining Additional Optical Disk Drives

You can copy existing optical disk drive definitions and modify them to create new ones by using the COPY line operator from the Optical Drive List panel. For more information on how to copy these existing optical disk drive definitions, see “Copying Optical Library and Drive Definitions” on page 376.

Maintaining and Modifying Optical Libraries and Optical Drives

After defining your optical libraries and optical disk drives to the SCDS and the optical configuration database, you may find that you need to change some of the definitions originally assigned to them. The following information provides options on how you can alter, copy, change, and delete definitions for optical libraries and optical disk drives.

Altering a 3995 Optical Library

You can use the optical library alter option to alter the attributes of an existing optical library. Altering a library results in updating the library definition within the specified SCDS and the attributes stored in the optical configuration database. The alter option is available only when the OAM address space is NOT active.

Figure 41 on page 332 is an example of the library alter panel for the 9246.

You can modify an optical library to change its definition in the optical configuration database by using either the 9246 LIBRARY ALTER panel, Figure 41 on page 332, or the 3995 LIBRARY ALTER panel, Figure 73 on page 369 and Figure 74 on page 371. To modify an optical library, start from the Library Application Selection panel, shown in Figure 54 on page 354, and specify the name of the SCDS containing the optical library you want to change. Specify the optical library name and select option **4**, ALTER. ISMF displays the appropriate library alter panel.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To ALTER Library, Specify:
  Description ==>
              ==>

  Model Number  . . . . .  133
  Base Device Number  . . 0940     (Valid base device number)
  Controlling Library . .          (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.          (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 73. 3995 Library Alter Panel (Page 1 of 2)

The following fields can be modified on the 3995 Library Alter panel (Page 1 of 2) to alter the 3995 library definition in the specified SCDS and in the optical configuration database for LIBRARY1.

MODEL NUMBER

Indicates the model number of the optical library. You can only change this field for 3995-1xx models. You cannot change the model number if the following conditions exist:

- The requested library is a PSEUDO library.
- The original library is one of the following model numbers: C3A, C1x, or C3x.

  Note: For simplicity, this publication refers to the following 3995 library models as C1x and C3x respectively: C12, C16, C18; C32, C34, C36, C38.
- The new model number specified is one of the following model numbers: C3A, SW3, SW4, C3x.

If you change the model number of a 3995-13x to that of another -13x model (for example, changing a model number from 131 to 133), you must perform the following steps:

1. Delete any existing operator-accessible drive from the SCDS **before** you change the model number if one had been previously defined as part of the 3995-13x optical library model.
2. Change the model number.
3. Redefine the operator-accessible drive by specifying the pseudo optical library name.
4. Repeat the above steps for other SCDSs that have the operator-accessible drive defined in them. These other SCDSs must have the same new model number so that they can be validated.

   Note: If the model 3995-13x optical library has a model 3995-11x optical library expansion unit connected to it before the MODEL NUMBER field is changed, it continues to be connected after the

MODEL NUMBER field is changed. The model number of the connected optical library expansion unit is also changed accordingly. For example, changing the model number from 131 to 132 also changes the model number of all attached drives in the library and expansion units.

If you change the MODEL NUMBER field from a -1xx to any other -1xx model (for example, changing the model number from a 111 to a 112), you must perform the following steps:

1. Change the optical library name in the CONTROLLING LIBRARY field to the appropriate corresponding -13x controlling library name.

   The system recalculates the base device number of the optical library and all the attached optical disk drives based on the base device number of the new controlling library specified.
2. Change the model number.

If you change the MODEL NUMBER field from a -13x model number to a -11x model number (for example, changing the model number from a 131 to a 111), you must perform the following steps:

1. Delete any existing operator-accessible drive from the SCDS that was previously defined as part of the 3995-13x optical library model.
2. Use the DELETE line operator to delete any connected -11x optical library **before** you change the model number.
3. Fill in the CONTROLLING LIBRARY field with the new controlling library name (3995-13x) to validate the controlling library.

   The system recalculates the base device number of the optical libraries and all of the attached optical disk drives based on the base device number of the new controlling library.

If you change the MODEL NUMBER field from a -11x model number to a -13x model number (for example, changing from a model number 113 to a 133), you must perform the following steps:

1. Fill in the BASE DEVICE NUMBER field with a valid 3995 device number.
2. Blank out the CONTROLLING LIBRARY NAME field.

   Note: If you are changing the model number to 131 or 132 from model 113, then the DEFAULT MEDIA TYPE field is blanked out. If you are changing the model number to 133 from model 111 or 112, then the DEFAULT MEDIA TYPE field is displayed as “3995”.

BASE DEVICE NUMBER

Changes to the base device number for a 3995 optical controlling library has the following results:

- All the base device numbers for all connected optical disk drives are recalculated based on the new base device number specified.
- All the base device numbers for any connected 3995 optical library expansion units are recalculated based on the new base device number specified.
- All the base device numbers for all the optical disk drives connected to the attached optical library expansion units are recalculated based on the new base device number specified.

Note: You cannot manually change the base device number for a 3995 optical library expansion unit model because this number is derived by the system based on the controlling library to which it is attached.

CONTROLLING LIBRARY

Changes to the CONTROLLING LIBRARY field for a connected 3995 library expansion unit model, has the following result:

- The new controlling library is verified. The library base device number of the connected 3995 optical library expansion unit and all of the attached optical disk drives are recalculated based on the base device number of the new 3995 controlling library model whose name is specified in the CONTROLLING LIBRARY field.

DEFAULT MEDIA TYPE

Changes to the DEFAULT MEDIA TYPE field restrict the type of optical media that can be used for the 3995 optical libraries with multifunction optical disk drives.

DEFAULT PSEUDO LIBRARY

Changes to the DEFAULT PSEUDO LIBRARY field may result in the volume being associated with a different pseudo library when it is ejected from this real library.

```
  Panel  Utilities  Scroll   Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    SYSTEM1  ===> YES  SYSTEM2  ===> NO   SYSTEM3  ===> NO   SYSTEM4  ===>
    SYSTEM5  ===>      SYSTEM6  ===>      SYSTEM7  ===>      SYSTEM8  ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>
             ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 74. 3995 Library Alter Panel (Page 2 of 2)

The following field can be modified on the 3995 Library Alter panel (Page 2 of 2) to alter the 3995 library system connection definition in the specified SCDS and in the optical configuration database for LIBRARY1.

INITIAL ONLINE STATUS

To change the system connectivity, or the online or offline status for the library, or both, alter the values on page 2 of 2 of the 3995 Library Alter panel (see Figure 74). A library can be defined as connected to any system that is defined in the SCDS, whether it was originally defined to that system or not. The library status is set to this value each time this SCDS is activated. The library must be connected to at least one system. Only one library can be online and connected (YES) to any system at a time. The default is blank. The following values can be used to specify initial online status:

- **YES** — online and connected to the system
- **NO** — offline and connected to the system
- **blank** — offline and not connected to the system.

Changing the 3995 Library Connectivity

The 3995 LIBRARY ALTER panels also allow the storage administrator to alter the library system connectivity, online or offline status, or both. This allows the library to be connected to a different system (different from the currently logged on system or different from the system on which the library was originally defined).

This change in system connectivity for the library allows customers to test OAM software and 3995 hardware on a test system without interrupting the actual production system; however, once a 3995 library is connected to a test system, it is not available to the production system until it is reconnected and the SCDS is reactivated. It should be noted that if the system connectivity of a 3995 optical library is changed, all the storage groups associated with that library must also have their system enablement status changed so they are enabled to the same system as the library. The test system then can use the same SMS configuration as the production system.

Note: DB2 databases belonging to OAM must be defined to the DB2 subsystem on the test system, because these databases cannot be shared between systems, and the optical configuration database used on the test system must be a copy of the one used on the production system.

To change the library's system connectivity and online status, assume a scenario where systems PRODSYS1-3 are part of an OAMplex on a production sysplex. Also assume that systems TESTSYS5-8 are being used as test systems and are not part of a separate test OAMplex on the same sysplex or a different sysplex.

Changing connectivity is done by changing the INITIAL ONLINE STATUS for system name PRODSYS1 to blank and the INITIAL ONLINE STATUS for system name TESTSYS5 to YES (see Figure 76 on page 374). LIBRARY1 is now online and connected to the TESTSYS5 system and can be used for testing. LIBRARY1 is now offline and not connected to any of the production systems (PRODSYS1-3) and is connected but offline to all the other test systems (TESTSYS6-8).

Changing the INITIAL ONLINE STATUS of a 3995 optical library dataserver controlling library model also updates the library definition in the specified SCDS to indicate the library is now connected to the new system. This also changes the system connectivity for any optical library expansion unit connected to these controlling libraries. In addition, all of the optical drive definitions in the specified SCDS for all the optical disk drives associated with this library are updated to indicate the optical drives are now connected to the new system.

Note: Changing the system connectivity of an optical library *does not* automatically change the system enablement status for the storage groups associated with that library. Therefore, the storage group definitions may need to be updated to provide the correct storage group enablement status. This allows the system connectivity of the optical library's associated storage groups' also to be changed from the OAMplex systems to the test system.

Attention: To reconnect the library to the original system (PRODSYS1), simply change the INITIAL ONLINE STATUS for system name PRODSYS1 back to **Y** (YES) and the system name TESTSYS1 back to **N** (NO) or **BLANK**. Also change the system enablement status of the storage groups which were changed from TESTSYS1 to PRODSYS1.

Indicate the library (LIBRARY1) for which the system connectivity and initial online status will change on the 3995 LIBRARY ALTER (Page 1 of 2) panel (Figure 75).

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER                Page 1 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

To ALTER Library, Specify:
  Description ==>
              ==>

  Model Number . . . . .  133
  Base Device Number  . . 0940     (Valid base device number)
  Controlling Library . .          (Library Name)        (if expansion unit)
  Default Media Type  . . 3995
  Default Pseudo Library.          (Pseudo Library name)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 75. 3995 Library Alter Panel (Page 1 of 2)

The 3995 LIBRARY ALTER (Page 2 of 2) panel (Figure 76 on page 374) is displayed next. This panel is used to indicate the initial online status and system connectivity of the library to various systems in the installation.

```
  Panel  Utilities  Scroll  Help
------------------------------------------------------------------------------
                             3995 LIBRARY ALTER               Page 2 of 2
Command ===>_

SCDS Name  . :SCDS.PRIMARY
Library Name :LIBRARY1
Library Type :REAL

  Initial Online Status (Yes, No, or Blank):
    PRODSYS1  ===>     PRODSYS2  ===>     PRODSYS3  ===>     PRODSYS4  ===>
    TESTSYS5  ===> YES TESTSYS6  ===> NO  TESTSYS7  ===> NO  TESTSYS8  ===> NO
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>
              ===>               ===>               ===>               ===>


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 76. 3995 Library Alter Panel (Page 2 of 2)

Confirming a System Mode Conversion

The Conversion Confirmation panel (see Figure 77) is displayed when there is an attempt to change the system mode of an SCDS from SYSTEMS(8) to SYSTEMS(32), or if the SCDS is in conflict with the current system mode for the active configuration. This panel occurs when the CDS currently being altered is in 8 system mode, and the SMS complex is in 32-system mode. For more information regarding system modes, refer to *z/OS DFSMSdfp Storage Administration Reference*.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                         CONVERSION CONFIRMATION PANEL
Command ===>

  To Confirm Conversion on the following CDS:

     CDS Name . . :SCDS.EIGHT.SYSTEM.MODE


  Specify the following:

      Perform Conversion:. . Y         (Y or N)



  This CDS is in 8-Name Mode, and the SMS complex is in 32-Name Mode.
  You must convert the CDS to 32-Name Mode prior to being able to
  access the CDS for update.  The conversion is PERMANENT.  A 32-Name
  Mode CDS cannot be converted to 8-Name Mode, be used by down-level
  systems or be used by DFSMS 1.3.0 systems running in 8-Name Mode.

Use ENTER to Perform Operation;
Use Help Command for Help; Use END Command to Exit.
```

Figure 77. Conversion Confirmation Panel

The following is a description of the fields on the Conversion Confirmation Panel:

CDS NAME
: Specifies the name of the CDS identified on the ISMF Optical Library Application Selection panel. This CDS system mode is in conflict with the system mode of the active configuration.

PERFORM CONVERSION
: Specifies a conversion should be done for the CDS to convert it to 32-system mode.

  Note: A CDS in 32-system mode cannot be converted to 8-system mode. Only the reverse is possible.

Altering an Optical Disk Drive

You can use the drive alter option to alter the attributes of an existing drive. Altering a drive definition results in updating the database drive row for that drive. The alter option is available only when OAM is **not** running.

To alter a drive definition in the optical configuration database and the SCDS, start from the Drive Application Selection panel, shown in Figure 54 on page 354, and specify the name of the SCDS containing the drive you want to change. Specify the drive name and select option **4**, ALTER. For the 3995 optical drive, ISMF displays the 3995 Drive Alter panel shown in Figure 78.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                              3995 DRIVE ALTER
Command ===>

SCDS Name   . . . . . :'SCDS.TEMP.PRIMARY'
Drive Name  . . . . . :P1D0
Drive Type  . . . . . :LIBRARY
Model Number  . . . . :132
Controlling Lib Name  :LIBRARY1

To ALTER Drive, Specify:

  Description ===> Line 1
              ===> Line 2
  Drive Number . . . 1


The Following Field may be Changed by Operator Accessible Drives Only:
  Library Name . . .                      (1 to 8 characters)

Use ENTER to Perform Verification;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 78. 3995 Drive Alter Panel

Notes:
1. You cannot change the drive number (always 5) for an operator-accessible drive (model 111, 112, or 113, the 3995 C-series models are exceptions).
2. For 3995 models 111, 112, 113, 131, 132, or 133, the drive number must be within the range of 1 to 4.
3. For all other 3995 models, the drive number must be within the range of 1 to 999.
4. You cannot change the library name for a drive unless it is an operator-accessible drive.

Copying Optical Library and Drive Definitions

You can copy existing optical drive and optical library definitions and modify them to create new ones using two methods. One method uses the attributes from the last optical disk drive or library definition. These values from the last definition are primed on the 3995 LIBRARY DEFINE or the 3995 DRIVE DEFINE panels, saving you from having to re-input similar data. It is simple to modify the attributes to define a new optical disk drive or optical library. After the attributes have been modified and you hit enter to finish the definition, the new optical disk drive or optical library is added to the SCDS and the optical configuration database. A second method of copying existing optical disk drives or optical library definitions to create new optical disk drives or optical libraries uses the COPY line operator from the OPTICAL LIBRARY LIST panel (for optical libraries) or the OPTICAL DRIVE LIST panel (for optical disk drives). To do so, enter the COPY line operator in the LINE OPERATOR column next to the optical disk drive or optical library you wish to copy. Press ENTER to copy the existing optical disk drive or optical library, and the COPY ENTRY panel will be displayed (see Figure 79).

Note: The copy function for 9246 and 3995 optical libraries operates in the same way, except that an optical controlling library may be required if you are copying an optical library expansion unit definition into a SCDS. The corresponding optical controlling library the SCDS is attached to must already be defined.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                           COPY ENTRY PANEL
Command ===>

  Definition will be copied from:

    Data Set Name . :'SCDS.TEMP.PRIMARY'
    Definition Name :LIB1
    Definition Type :OPTICAL LIBRARY

  Specify "Copy To" Definition:

    Data Set Name . .   'SCDS.TEMP.PRIMARY'
                                    (1 to 46 characters)

    Definition Name . .             (1 to 8 characters, fully specified)

    Enter "/" to select option    _  Perform Alter



 Use ENTER to Perform COPY;
 Use HELP Command for HELP; Use END command to Exit.
```

Figure 79. Copy Entry Panel

The *from* DATA SET NAME field identifies the source for the copy. It is primed with the value you specified on the Optical Drive (or Library) Application Selection panel. The *from* DEFINITION NAME field identifies the name of the optical disk drive or optical library to be copied. This field is primed with the value from the DRIVE NAME or LIBRARY NAME field of the Optical Drive (or Library) List panel.

The *to* DATA SET NAME field identifies the target SCDS of the copy. It must be a name of an SCDS. It is primed with the value of the *from* DATA SET NAME if the *from* DATA SET NAME contains an SCDS name. It is primed with blanks if the *from*

DATA SET NAME is 'ACTIVE'. The *to* DEFINITION NAME field identifies the name of the optical disk drive or optical library. It is primed with blanks.

In the PERFORM ALTER field, indicate if you want to change some of the attributes of the source copy. If you specify **Y** (YES), the appropriate Alter panel is displayed. If you specify **N** (NO), you remain on the Copy Entry panel, where you can perform another copy or return to the original List panel.

When copying an optical disk drive or optical library definition from one SCDS into another SCDS, you do not need to select the PERFORM ALTER option. In the case where an optical disk drive or optical library definition is copied within the same SCDS, you must choose the PERFORM ALTER option because optical disk drives and optical libraries must differ from one definition to another. (In particular, those fields that are used in addressing an optical disk drive or optical library, such as CTC device number and SCSI address, must be unique.)

When you have specified the values, press ENTER to perform the copy.

Deleting an Optical Library

You can delete an optical library definition within the specified SCDS. Before the optical library definition is deleted, all optical disk drives defined for that optical library are deleted and all storage groups constructs that reference the optical library are updated to not reference that optical library. This is done automatically as part of the optical library deletion process.

From the Optical Library List panel, enter DELETE in the LINE OPERATOR column next to the optical library you wish to delete. When you press ENTER, the Confirm Delete Request panel, shown in Figure 80, is displayed.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            CONFIRM DELETE REQUEST
Command ===>

  To Confirm Deletion on the following Optical Library:

     Optical Library Name :LIB1
     Residing in SCDS . . :'SCDS.TEMP.PRIMARY'

  Specify the following:
      Enter "/" to select option   _ Perform deletion

  Note:If Deletion is Performed, All Drive Definitions associated with the
        Library will be Deleted and all Storage Group Constructs that
        reference the Library will be Updated.  In addition, if the Library is
        a 3995 Model 131, 132, 133 or C3A, then any Library connected to it
        (such as a 3995 Model 111, 112, 113, C12, C16, C18, C32, C34, C36
        or C38) will also be deleted in the same manner.  Definitions
        will not be removed from the Optical Configuration Database
        DB2 tables.  DB2 (SPUFI) can be used for this purpose.

Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 80. Deleting an Optical Library

Confirm that the displayed library is the one that you want to delete. If it is, enter **Y** for YES and press ENTER. The Optical Library List should appear with '*DELETE' in the LINE OPERATOR column next to the deleted library.

Note: If an optical controlling library is deleted, any attached optical library expansion unit and optical disk drives are also deleted. To delete an optical library from the optical configuration database, use the DB2I or SPUFI tools.

Deleting an Optical Disk Drive

You can delete an optical disk drive definition within the specified SCDS. From the Optical Drive List panel, enter **DELETE** in the LINE OPERATOR column next to the optical disk drive you wish to delete. When you press ENTER, the Confirm Delete Request panel, shown in Figure 81, is displayed. To delete an optical disk drive from the optical configuration database, use the DB2I or SPUFI tools.

```
  Panel  Utilities  Help
------------------------------------------------------------------------------
                            CONFIRM DELETE REQUEST
Command ===>

  To Confirm Deletion on the Following Optical Drive:

    Optical Drive Name :LIB0
    Residing in SCDS . :'SCDS.TEMP.PRIMARY'



  Specify the Following:
      Enter "/" to select option   _ Perform Deletion



  Note: If deletion is performed, the drive definition will not be removed
         from the Optical Configuration Database DB2 tables.



Use ENTER to Perform Operation;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 81. Deleting an Optical Disk Drive

Confirm that the displayed drive is the one that you want to delete. If it is, enter **Y** for YES and press ENTER. The Optical Drive List appears with ***DELETE** in the LINE OPERATOR column next to the deleted drive.

Deleting a drive has no effect on the optical configuration database. When you delete a drive, the definition for that drive is removed from the specified SCDS. When the SCDS is reactivated, the deleted drive is unavailable to the system. To delete a drive from the optical configuration database, use the DB2I or SPUFI tools. Ensure that the other SCDSs that reference the deleted drive are updated; otherwise, OAM initialization will fail.

Using DELETE FORCE to Delete an Optical Library or Optical Drive

Under certain circumstances, the ISMF Library Management **DELETE** line operator will not work. To get around this problem the **FORCE** parameter is used. To cause deletion of an optical library or optical disk drive when the standard **DELETE** line operator does not work, enter **DELETE FORCE**.

Remember, deleting an optical library or optical drive has no effect on the optical configuration database. Instead, when you delete an optical library or an optical disk drive, the definition for that deleted device is unavailable to the system when the SCDS is reactivated. Therefore, once the record in the SCDS is removed, you can redefine the record using the values found in the optical configuration database. Make sure you save the names of the optical libraries, optical disk drives, or both so they can be redefined.

Attention: **DELETE FORCE** will *not* delete any other entries in the SCDS. For example, if a library is deleted using the normal **DELETE** line operator, any optical disk drives associated with the optical library will also be deleted. However, if the optical library is deleted using **DELETE FORCE**, the optical disk drives associated with the optical library will not be deleted.

Appendix B. Sample Library Members

This appendix provides examples of some of the SAMPLIB members that enable you to install and use OAM support. The SAMPLIB members provided include the following examples:

SAMPLIB Member and Page Reference
“Changing System Libraries” • “CBRAPROC” on page 382 • “CBRIPROC” on page 382
“Creating Object Databases” on page 383 • “CBRIALC0” on page 383 • “CBRIALCX” on page 386 • “CBRIALCY” on page 389 • “CBRISQL0” on page 390 • “CBRISQLX” on page 394 • “CBRISQLY” on page 397
“OAM Optical Configuration Database” on page 399 • “CBRSAMPL” on page 399 • “CBRSMB2” on page 407 • “CBRSM150” on page 409 • “CBRSMR13” on page 411 • “CBRSMERG” on page 415 • “CBRSG100” on page 422
“Application Plans” on page 432 (Only text descriptions are provided; no examples are available.)
“OAM Installation Verification Program and OAMUTIL” on page 433 • “OAMIVP” on page 433 • “CBRSAMUT” on page 434
“Automatic Class Selection” on page 436 • “CBRHSC” on page 437 • “CBRHMC” on page 441 • “CBRHSG” on page 449

OAM processing is dependent on the DB2 tables that are created by the sample jobs. It is crucial that the tables are created, and migrated where applicable, exactly as specified in the sample jobs without change. Changes to columns or other characteristics of these tables may cause errors initializing or using OAM. These tables are not intended as an interface. Though accessible through system administration authority, access should be restricted from end users.

Changing System Libraries

SAMPLIB members CBRAPROC and CBRIPROC are supplied to create the OAM and OTIS procedures in PROCLIB.

CBRAPROC

SAMPLIB member CBRAPROC, as shown in Figure 82, creates member OAM in PROCLIB.

```
//CBRAPROC JOB  MSGLEVEL=(1,1),MSGCLASS=A
//********************************************************************
//*
//*  $SEG(CBRAPROC) COMP(OSMC) PROD(OAM):
//*
//*  OAM Update PROCLIB Job (for OAM procedure).
//*
//*  This job will create a procedure in PROCLIB that can be used
//*  to start OAM.
//*
//*  NOTE:If the DB2 load module library containing DSNALI is
//*        not in the LNKLST concatenation, either include
//*        the DB2 load module library in the SYS1.LINKLIB
//*        concatenation (LNKLSTxx) or add a STEPLIB DD to
//*        this PROCEDURE.
//*
//*        If a STEPLIB is used, then that concatenation must be
//*        APF-authorized.
//*
//********************************************************************
//         EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD   SYSOUT=A
//SYSUT2   DD   DSNAME=SYS1.PROCLIB,DISP=SHR
//SYSIN    DD   DATA
./         ADD   NAME=OAM,LEVEL=01,SOURCE=0,LIST=ALL
./      NUMBER   NEW1=10,INCR=10
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,RESTART=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
// PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD,EJECT=&EJECT')
//SYSABEND DD SYSOUT=A
./      ENDUP
```

Figure 82. CBRAPROC SAMPLIB Member

CBRIPROC

SAMPLIB member CBRIPROC, as shown in Figure 83 on page 383, creates member OTIS in PROCLIB.

```
//CBRIPROC JOB  MSGLEVEL=(1,1),MSGCLASS=A
//*******************************************************************
//*
//*  $SEG(CBRIPROC) COMP(OSR) PROD(OAM):
//*
//*  OAM Update PROCLIB Job (for OTIS procedure).
//*
//*  This job will create a procedure in PROCLIB that can be used
//*  to start OTIS.
//*
//*  NOTE:If the DB2 load module library containing DSNALI is
//*       not in the LNKLST concatenation, either include
//*       the DB2 load module library in the SYS1.LINKLIB
//*       concatenation (LNKLSTxx) or add a STEPLIB DD to
//*       this PROCEDURE.
//*
//*       If a STEPLIB is used, then that concatenation must be
//*       APF-authorized.
//*
//*******************************************************************
//STEP1    EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD   SYSOUT=A
//SYSUT2   DD   DSNAME=SYS1.PROCLIB,DISP=SHR
//SYSIN    DD   DATA
./        ADD   NAME=OTIS,LEVEL=01,SOURCE=0,LIST=ALL
./      NUMBER   NEW1=10,INCR=10
//OTIS    PROC
//IEFPROC EXEC PGM=CBRIIAS,REGION=0M
//SYSABEND DD SYSOUT=A
./      ENDUP
```

Figure 83. CBRIPROC SAMPLIB Member

Creating Object Databases

To create the object databases for OAM, several jobs are supplied as members in SAMPLIB. Three members contain the data set allocation jobs and three members contain the DB2 database definition jobs. The CBRIALC0 job allocates the VSAM data sets for the DB2 object storage databases, and the CBRIALCX and CBRIALCY jobs allocate the VSAM data sets for the DB2 object administration database. Similarly, the CBRISQL0 database definition job defines the object storage databases, and the CBRISQLX and CBRISQLY jobs define the object administration database.

For each database used, the corresponding allocation and database definition job steps must be run successfully. Before the jobs are run, they must be modified for your installation. Refer to the job prologs for the required modifications and related information.

CBRIALC0

SAMPLIB member CBRIALC0, as shown in Figure 84 on page 384, provides data set allocation for the OAM object tables and directories. This job must be modified and run successfully before OAM is used.

```
//CBRIALC0 JOB  MSGLEVEL=(1,1),MSGCLASS=A
//*************************************************************
//*
//*  $SEG(CBRIALC0) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Data Set Allocation Job (for Object Tables
//*  and Directories).
//*
//*  Run CBRIALC0 to define a VSAM ESDS that will be
//*  used by DB2 for an OSR object database.
//*
//*  Prior to executing this job you need to make the
//*  following modifications:
//*
//*  1. Change "vol_ser" to the volume serials that your
//*     target database should reside on.
//*  2. Change "pri_alloc" and "sec_alloc" to the desired
//*     number of cylinders for each particular VSAM ESDS
//*     being defined. For example, CYLINDER(pri_alloc
//*     sec_alloc) may be CYLINDER(200 10).
//*  3. Change "cat_name" to the name of the catalog you
//*     will be using under DB2.
//*  4. If you intend on using the DSN1COPY utility to copy
//*     these data bases, then you must include the REUSE keyword
//*     in the DEFINE CLUSTER command for each data base.
//*  5. Change "osg_hlq" to the high level qualifier to be used
//*     for the object storage group definition and tables.
//*     This is the qualifier used on the object storage group
//*     define through ISMF and used by OAM and OSR for all access
//*     to the object storage group's directories and data tables.
//*  6. Add additional job steps, repeating all statements in the
//*     first STEP01, for each object storage group defined in your
//*     configuration.  In each repeated step, change the qualifier
//*     to match the qualifier for each object storage group.
//*
//*  Following data set allocations, run CBRISQL0 (provided
//*  in SAMPLIB for your modification) to define
//*  DB2 databases, table spaces, indexes, views, etc.
//*  using the data sets allocated by this job.
//*
//*  If you have run this job and want to start over
//*  again, just issue a DROP for each database that was
//*  previously defined in DB2.
//*
//*************************************************************
//STEP01  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 DELETE                                                 -
 cat_name.DSNDBC.osg_hlq.OSMDTS.I0001.A001              -
     CLUSTER                                            -
     PURGE
 DELETE                                                 -
 cat_name.DSNDBC.osg_hlq.OSMOTS04.I0001.A001            -
     CLUSTER                                            -
     PURGE
 DELETE                                                 -
 cat_name.DSNDBC.osg_hlq.OSMOTS32.I0001.A001            -
     CLUSTER                                            -
     PURGE
 SET LASTCC=0
 SET MAXCC=0
```

Figure 84. CBRIALC0 SAMPLIB Member (Part 1 of 3)

```
DEFINE CLUSTER                                       -
(NAME(cat_name.DSNDBC.osg_hlq.OSMDTS.I0001.A001)     -
        LINEAR                                       -
        SHAREOPTIONS(3 3)                            -
        VOLUMES(vol_ser)                             -
        CYLINDERS(pri_alloc sec_alloc)               -
        UNIQUE   )                                   -
      DATA                                           -
(NAME(cat_name.DSNDBD.osg_hlq.OSMDTS.I0001.A001))
DEFINE CLUSTER                                       -
(NAME(cat_name.DSNDBC.osg_hlq.OSMOTS04.I0001.A001)   -
        LINEAR                                       -
        SHAREOPTIONS(3 3)                            -
        VOLUMES(vol_ser)                             -
        CYLINDERS(pri_alloc sec_alloc)               -
        UNIQUE   )                                   -
      DATA                                           -
(NAME(cat_name.DSNDBD.osg_hlq.OSMOTS04.I0001.A001))
DEFINE CLUSTER                                       -
(NAME(cat_name.DSNDBC.osg_hlq.OSMOTS32.I0001.A001)   -
        LINEAR                                       -
        SHAREOPTIONS(3 3)                            -
        VOLUMES(vol_ser)                             -
        CYLINDERS(pri_alloc sec_alloc)               -
        UNIQUE   )                                   -
      DATA                                           -
(NAME(cat_name.DSNDBD.osg_hlq.OSMOTS32.I0001.A001))
DELETE                                               -
cat_name.DSNDBC.osg_hlq.OBJDIRX1.I0001.A001          -
    CLUSTER                                          -
    PURGE
DELETE                                               -
cat_name.DSNDBC.osg_hlq.OBJDIRX2.I0001.A001          -
    CLUSTER                                          -
    PURGE
DELETE                                               -
cat_name.DSNDBC.osg_hlq.OBJDIRX3.I0001.A001          -
    CLUSTER                                          -
    PURGE
DELETE                                               -
cat_name.DSNDBC.osg_hlq.OBJT04X1.I0001.A001          -
    CLUSTER                                          -
    PURGE
DELETE                                               -
cat_name.DSNDBC.osg_hlq.OBJT32X1.I0001.A001          -
    CLUSTER                                          -
    PURGE
SET LASTCC=0
SET MAXCC=0
DEFINE CLUSTER                                       -
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX1.I0001.A001)   -
        LINEAR                                       -
        SHAREOPTIONS(3 3)                            -
        VOLUMES(vol_ser)                             -
        CYLINDERS(pri_alloc sec_alloc)               -
        UNIQUE   )                                   -
      DATA                                           -
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX1.I0001.A001))
```

Figure 84. CBRIALC0 SAMPLIB Member (Part 2 of 3)

```
DEFINE CLUSTER                                             -
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX2.I0001.A001)   -
        LINEAR                                             -
        SHAREOPTIONS(3 3)                                  -
        VOLUMES(vol_ser)                                   -
        CYLINDERS(pri_alloc sec_alloc)                     -
        UNIQUE   )                                         -
      DATA                                                 -
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX2.I0001.A001))
DEFINE CLUSTER                                             -
(NAME(cat_name.DSNDBC.osg_hlq.OBJDIRX3.I0001.A001)   -
        LINEAR                                             -
        SHAREOPTIONS(3 3)                                  -
        VOLUMES(vol_ser)                                   -
        CYLINDERS(pri_alloc sec_alloc)                     -
        UNIQUE   )                                         -
      DATA                                                 -
(NAME(cat_name.DSNDBD.osg_hlq.OBJDIRX3.I0001.A001))
DEFINE CLUSTER                                             -
(NAME(cat_name.DSNDBC.osg_hlq.OBJT04X1.I0001.A001)   -
        LINEAR                                             -
        SHAREOPTIONS(3 3)                                  -
        VOLUMES(vol_ser)                                   -
        CYLINDERS(pri_alloc sec_alloc)                     -
        UNIQUE   )                                         -
      DATA                                                 -
(NAME(cat_name.DSNDBD.osg_hlq.OBJT04X1.I0001.A001))
DEFINE CLUSTER                                             -
(NAME(cat_name.DSNDBC.osg_hlq.OBJT32X1.I0001.A001)   -
        LINEAR                                             -
        SHAREOPTIONS(3 3)                                  -
        VOLUMES(vol_ser)                                   -
        CYLINDERS(pri_alloc sec_alloc)                     -
        UNIQUE   )                                         -
      DATA                                                 -
(NAME(cat_name.DSNDBD.osg_hlq.OBJT32X1.I0001.A001))
/*
```

Figure 84. CBRIALC0 SAMPLIB Member (Part 3 of 3)

CBRIALCX

SAMPLIB member CBRIALCX, as shown in Figure 85 on page 387, provides data set allocation for part of the OAM administration tables. This job must be modified and run successfully before OAM is used.

```
//CBRIALCX JOB  MSGLEVEL=(1,1),MSGCLASS=A
//*********************************************************
//*
//*  $SEG(CBRIALCX) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Data Set Allocation Job (for Administration
//*  Databases).
//*
//*  Run CBRIALCX to define the VSAM ESDSs that
//*  will be used by DB2 for the OAMADMIN databases
//*  required by OAM.
//*
//*  Prior to executing this job you need to make the
//*  following modifications:
//*
//*  1. Change "vol_ser" to the volume serials that your
//*     target database should reside on.
//*  2. Change "pri_alloc" and "sec_alloc" to the desired
//*     number of cylinders for each particular VSAM ESDS
//*     being defined. For example, CYLINDER(pri_alloc
//*     sec_alloc) may be CYLINDER(200 10).
//*  3. Change "cat_name" to the name of the catalog you
//*     will be using under DB2.
//*
//*  Following data set allocations, run CBRISQLX (provided
//*  in SAMPLIB for your modification) to define the
//*  OAM Administration databases using the data sets
//*  allocated by this job.
//*
//*  If you have run this job and want to start over
//*  again, just issue a DROP for each OSR database
//*  in DB2 that was previously defined by this job.
//*
//*********************************************************
//STEP0X  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 DELETE                                                 -
 cat_name.DSNDBC.OAMADMIN.MCIND.I0001.A001              -
     CLUSTER                                            -
     PURGE
 DELETE                                                 -
 cat_name.DSNDBC.OAMADMIN.SCIND.I0001.A001              -
     CLUSTER                                            -
     PURGE
 DELETE                                                 -
 cat_name.DSNDBC.OAMADMIN.COLIND.I0001.A001             -
     CLUSTER                                            -
     PURGE
 SET LASTCC=0
 SET MAXCC=0
 DEFINE CLUSTER                                         -
 (NAME(cat_name.DSNDBC.OAMADMIN.MCIND.I0001.A001)       -
         LINEAR                                         -
         SHAREOPTIONS(3 3)                              -
         VOLUMES(vol_ser)                               -
         CYLINDERS(pri_alloc sec_alloc)                 -
         UNIQUE   )                                     -
       DATA                                             -
 (NAME(cat_name.DSNDBD.OAMADMIN.MCIND.I0001.A001))
```

Figure 85. CBRIALCX SAMPLIB Member (Part 1 of 3)

```
DEFINE CLUSTER                                        -
(NAME(cat_name.DSNDBC.OAMADMIN.SCIND.I0001.A001)      -
        LINEAR                                        -
        SHAREOPTIONS(3 3)                             -
        VOLUMES(vol_ser)                              -
        CYLINDERS(pri_alloc sec_alloc)                -
        UNIQUE   )                                    -
      DATA                                            -
(NAME(cat_name.DSNDBD.OAMADMIN.SCIND.I0001.A001))
DEFINE CLUSTER                                        -
(NAME(cat_name.DSNDBC.OAMADMIN.COLIND.I0001.A001)     -
        LINEAR                                        -
        SHAREOPTIONS(3 3)                             -
        VOLUMES(vol_ser)                              -
        CYLINDERS(pri_alloc sec_alloc)                -
        UNIQUE   )                                    -
      DATA                                            -
(NAME(cat_name.DSNDBD.OAMADMIN.COLIND.I0001.A001))
DELETE                                                -
cat_name.DSNDBC.OAMADMIN.CBRMGTX.I0001.A001           -
    CLUSTER                                           -
    PURGE
DELETE                                                -
cat_name.DSNDBC.OAMADMIN.CBRSTOX.I0001.A001           -
    CLUSTER                                           -
    PURGE
DELETE                                                -
cat_name.DSNDBC.OAMADMIN.CBRCLTX1.I0001.A001          -
    CLUSTER                                           -
    PURGE
DELETE                                                -
cat_name.DSNDBC.OAMADMIN.CBRCLTX2.I0001.A001          -
    CLUSTER                                           -
    PURGE
DELETE                                                -
cat_name.DSNDBC.OAMADMIN.CBRCLTX3.I0001.A001          -
    CLUSTER                                           -
    PURGE
SET LASTCC=0
SET MAXCC=0
DEFINE CLUSTER                                        -
(NAME(cat_name.DSNDBC.OAMADMIN.CBRMGTX.I0001.A001)    -
        LINEAR                                        -
        SHAREOPTIONS(3 3)                             -
        VOLUMES(vol_ser)                              -
        CYLINDERS(pri_alloc sec_alloc)                -
        UNIQUE   )                                    -
      DATA                                            -
(NAME(cat_name.DSNDBD.OAMADMIN.CBRMGTX.I0001.A001))
DEFINE CLUSTER                                        -
(NAME(cat_name.DSNDBC.OAMADMIN.CBRSTOX.I0001.A001)    -
        LINEAR                                        -
        SHAREOPTIONS(3 3)                             -
        VOLUMES(vol_ser)                              -
        CYLINDERS(pri_alloc sec_alloc)                -
        UNIQUE   )                                    -
      DATA                                            -
(NAME(cat_name.DSNDBD.OAMADMIN.CBRSTOX.I0001.A001))
DEFINE CLUSTER                                        -
(NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX1.I0001.A001)   -
        LINEAR                                        -
        SHAREOPTIONS(3 3)                             -
        VOLUMES(vol_ser)                              -
        CYLINDERS(pri_alloc sec_alloc)                -
        UNIQUE   )                                    -
      DATA                                            -
(NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX1.I0001.A001))
```

Figure 85. CBRIALCX SAMPLIB Member (Part 2 of 3)

```
DEFINE CLUSTER                                          -
(NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX2.I0001.A001)  -
        LINEAR                                          -
        SHAREOPTIONS(3 3)                               -
        VOLUMES(vol_ser)                                -
        CYLINDERS(pri_alloc sec_alloc)                  -
        UNIQUE   )                                      -
      DATA                                              -
(NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX2.I0001.A001))
DEFINE CLUSTER                                          -
(NAME(cat_name.DSNDBC.OAMADMIN.CBRCLTX3.I0001.A001)  -
        LINEAR                                          -
        SHAREOPTIONS(3 3)                               -
        VOLUMES(vol_ser)                                -
        CYLINDERS(pri_alloc sec_alloc)                  -
        UNIQUE   )                                      -
      DATA                                              -
(NAME(cat_name.DSNDBD.OAMADMIN.CBRCLTX3.I0001.A001))
```

Figure 85. CBRIALCX SAMPLIB Member (Part 3 of 3)

CBRIALCY

SAMPLIB member CBRIALCY, as shown in Figure 86, provides data set allocation for part of the OAM object tables and directories. This job must be modified and run successfully before OAM is used.

```
//CBRIALCY JOB  MSGLEVEL=(1,1),MSGCLASS=A
//*************************************************************
//*
//*  $SEG(CBRIALCY) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Data Set Allocation Job (for Administration
//*  Databases).
//*
//*  Run CBRIALCY to define the VSAM ESDSs that
//*  will be used by DB2 for the OAMADMIN databases
//*  required by OAM.
//*
//*  Prior to executing this job you need to make the
//*  following modifications:
//*
//*  1. Change "vol_ser" to the volume serials that your
//*     target database should reside on.
//*  2. Change "pri_alloc" and "sec_alloc" to the desired
//*     number of cylinders for each particular VSAM ESDS
//*     being defined. For example, CYLINDER(pri_alloc
//*     sec_alloc) may be CYLINDER(200 10).
//*  3. Change "cat_name" to the name of the catalog you
//*     will be using under DB2.
//*
//*  Following data set allocations, run CBRISQLY (provided
//*  in SAMPLIB for your modification) to define the
//*  OAM Administration databases using the data sets
//*  allocated by this job.
//*
```

Figure 86. CBRIALCY SAMPLIB Member (Part 1 of 2)

```
//*************************************************************
//STEP0X  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
 DELETE                                                   -
 cat_name.DSNDBC.OAMADMIN.CBRMGTY.I0001.A001              -
     CLUSTER                                              -
     PURGE
 DELETE                                                   -
 cat_name.DSNDBC.OAMADMIN.CBRSTOY.I0001.A001              -
     CLUSTER                                              -
     PURGE
 DEFINE CLUSTER                                           -
 (NAME(cat_name.DSNDBC.OAMADMIN.CBRMGTY.I0001.A001)       -
         LINEAR                                           -
         SHAREOPTIONS(3 3)                                -
         VOLUMES(vol_ser)                                 -
         CYLINDERS(pri_alloc sec_alloc)                   -
         UNIQUE   )                                       -
       DATA                                               -
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRMGTY.I0001.A001))
 DEFINE CLUSTER                                           -
 (NAME(cat_name.DSNDBC.OAMADMIN.CBRSTOY.I0001.A001)       -
         LINEAR                                           -
         SHAREOPTIONS(3 3)                                -
         VOLUMES(vol_ser)                                 -
         CYLINDERS(pri_alloc sec_alloc)                   -
         UNIQUE   )                                       -
       DATA                                               -
 (NAME(cat_name.DSNDBD.OAMADMIN.CBRSTOY.I0001.A001))
```

Figure 86. CBRIALCY SAMPLIB Member (Part 2 of 2)

CBRISQL0

SAMPLIB member CBRISQL0, as shown in Figure 87, provides the DB2 definitions for the OAM object tables and directories. This job must be modified and run successfully before OAM is used.

```
//CBRISQL0 JOB  MSGLEVEL=(1,1),MSGCLASS=A
//****************************************************************
//*
//*  $SEG(CBRISQL0) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Database Definition Job (for Object Tables
//*  and Directories).
//*
//*  This job will create an OSR database, table, and
//*  index in DB2 for an object storage group.
//*
```

Figure 87. CBRISQL0 SAMPLIB Member (Part 1 of 5)

```
//*  Before running this job, you must change the following:
//*
//*  1. Change "cat_name" to the DB2 VCAT name used
//*     for defining the VSAM data sets in CBRIALC0.
//*
//*  2. Change "auth_id" to the identifier(s)
//*     authorized for the respective group.
//*
//*  3. Change the name in the DSN SYSTEM(DB2) statement to
//*     the name of the DB2 Subsystem in your installation.
//*
//*  4. Change the data set name in the RUN statement
//*     LIB('DB2.RUNLIB.LOAD') phrase to the data set name used
//*     in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  5. Change the PLAN name (DSNTIA21) in the RUN statement to
//*     match your current DB2 version and release level.
//*
//*  6. If you plan to use just one collection, reverse the
//*     order of ODCLID and ODPENDDT in index OBJDIRX2.
//*
//*  7. Change "osg_hlq" to the high level qualifier to be used
//*     for the object storage group definition and tables.
//*     This is the qualifier used on the object storage group
//*     define through ISMF and used by OAM and OSR for all access
//*     to the object storage group's directories and data tables.
//*
//*  8. Add additional job steps, repeating all statements in
//*     STEP01, for each object storage group defined in your
//*     configuration.  In each repeated step, change the qualifier
//*     to match the qualifier for each object storage group.
//*
//*  CHANGE ACTIVITY:
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release
//*    $01=OY26530 320 891113 TUCTNN: Removed OTSEG from OBJT04X1
//*    $02=OY33596 320 901019 TUCHTT: Changed index OBJDIRX1 to
//*                                   ODCREATS. Changed index
//*                                   OBJDIRX2 to ODPENDDT, ODCLID,
//*                                   ODCREATS. Changed index
//*                                   OBJDIRX3 to ODNAME, ODCLID.
//*    $L1=JDP3331 331 910614 TUCKSG: Reverse order of ODCLID
//*                                   and ODPENDDT for index
//*                                   OBJDIRX2.
//*    $L2=HDZ11D0 140 970331 TUCSPP: Specify TYPE 1 INDEX for
//*                                   DB2 4.1 or above level
//*    $L3=HDZ11E0 150 970812 TUCLJT: GROUP00-GROUP99 qualifier
//*                                   restriction removed.  Single
//*                                   set of JCL provided and user
//*                                   to customize to installation.
//*    $L4=HDZ11G0 R13 001012 TUCLJT: Add ODBK2LOC and ODBK2SEC for
//*                                   Multiple Object Backup Support
//*                                   Also:
//*                                   - Removed reference to type 1
//*                                     indexes, which are no longer
//*                                     supported by DB2
//*                                   - Removed SUBPAGES from CREATE
//*                                     statements, since they are
//*                                     only for type 1 indexes
```

Figure 87. CBRISQL0 SAMPLIB Member (Part 2 of 5)

```
//*
//*****************************************************************
//STEP00  EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA21) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
          CREATE DATABASE   osg_hlq;
          COMMIT;
          CREATE TABLESPACE OSMDTS
                 IN         osg_hlq
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 SEGSIZE    64
                 BUFFERPOOL BP0;
          CREATE TABLESPACE OSMOTS04
                 IN         osg_hlq
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 SEGSIZE    64
                 BUFFERPOOL BP2;
          CREATE TABLESPACE OSMOTS32
                 IN         osg_hlq
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 SEGSIZE    64
                 BUFFERPOOL BP32K;
          COMMIT;
          CREATE TABLE osg_hlq.OSM_OBJ_DIR
                 (
                  ODVER    CHAR(1)      NOT NULL,
                  ODSIZE   INTEGER      NOT NULL,
                  ODCREATS TIMESTAMP    NOT NULL,
                  ODEXPDT  DATE         NOT NULL,
                  ODLREFDT DATE         NOT NULL,
                  ODPENDDT DATE         NOT NULL,
                  ODMCASDT DATE         NOT NULL,
                  ODSCNUM  SMALLINT     NOT NULL,
                  ODMCNUM  SMALLINT     NOT NULL,
                  ODLOCFL  CHAR(1)      NOT NULL,
                  ODLSLOC  CHAR(6)      NOT NULL,
                  ODSECLOC INTEGER      NOT NULL,
                  ODBKLOC  CHAR(6)      NOT NULL,
                  ODBKSEC  INTEGER      NOT NULL,
                  ODCLID   INTEGER      NOT NULL,
                  ODNAME   VARCHAR(44)  NOT NULL
                  ODBK2LOC  CHAR(6)     NOT NULL,
                  ODBK2SEC  INTEGER     NOT NULL,
                 )
                 IN osg_hlq.OSMDTS;
          CREATE UNIQUE INDEX osg_hlq.OBJDIRX1
                 ON osg_hlq.OSM_OBJ_DIR
                 (
                  ODCREATS ASC
                 )
                 CLUSTER
                 USING      VCAT cat_name
                 CLOSE      NO
                 BUFFERPOOL BP1
                 PCTFREE    10;
```

Figure 87. CBRISQL0 SAMPLIB Member (Part 3 of 5)

```
CREATE UNIQUE INDEX osg_hlq.OBJDIRX2
       ON osg_hlq.OSM_OBJ_DIR
       (
        ODCLID   ASC,
        ODPENDDT ASC,
        ODCREATS ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       BUFFERPOOL BP1
       PCTFREE    10;
CREATE UNIQUE INDEX osg_hlq.OBJDIRX3
       ON osg_hlq.OSM_OBJ_DIR
       (
        ODNAME   ASC,
        ODCLID   ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       BUFFERPOOL BP1
       PCTFREE    10;
COMMIT;
CREATE TABLE osg_hlq.OSM_04K_OBJ_TBL
       (
        OTVER    CHAR(1)      NOT NULL,
        OTSEG    SMALLINT     NOT NULL,
        OTCLID   INTEGER      NOT NULL,
        OTNAME   VARCHAR(44)  NOT NULL,
        OTOBJ    LONG VARCHAR NOT NULL
       )
       IN osg_hlq.OSMOTS04;
CREATE UNIQUE INDEX osg_hlq.OBJT04X1
       ON osg_hlq.OSM_04K_OBJ_TBL
       (
        OTCLID   ASC,
        OTNAME   ASC
       )
       CLUSTER
       USING      VCAT cat_name
       CLOSE      NO
       BUFFERPOOL BP1
       PCTFREE    10;
CREATE TABLE osg_hlq.OSM_32K_OBJ_TBL
       (
        OTVER    CHAR(1)      NOT NULL,
        OTSEG    SMALLINT     NOT NULL,
        OTCLID   INTEGER      NOT NULL,
        OTNAME   VARCHAR(44)  NOT NULL,
        OTOBJ    LONG VARCHAR NOT NULL
       )
       IN osg_hlq.OSMOTS32;
CREATE UNIQUE INDEX osg_hlq.OBJT32X1
       ON osg_hlq.OSM_32K_OBJ_TBL
       (
        OTCLID   ASC,
        OTNAME   ASC,
        OTSEG    ASC
       )
       CLUSTER
       USING      VCAT cat_name
       CLOSE      NO
       BUFFERPOOL BP1
       PCTFREE    10;
```

Figure 87. CBRISQL0 SAMPLIB Member (Part 4 of 5)

```
COMMIT;
CREATE VIEW
       osg_hlq.V_OSM_OBJ_DIR
     AS SELECT ALL * FROM
       osg_hlq.OSM_OBJ_DIR;
CREATE VIEW
       osg_hlq.V_OSM_04K_OBJ_TBL
     AS SELECT ALL * FROM
       osg_hlq.OSM_04K_OBJ_TBL;
CREATE VIEW
       osg_hlq.V_OSM_32K_OBJ_TBL
     AS SELECT ALL * FROM
       osg_hlq.OSM_32K_OBJ_TBL;
GRANT ALL ON
       osg_hlq.V_OSM_OBJ_DIR
     TO auth_id;
GRANT ALL ON
       osg_hlq.V_OSM_04K_OBJ_TBL
     TO auth_id;
GRANT ALL ON
       osg_hlq.V_OSM_32K_OBJ_TBL
     TO auth_id;
COMMIT;
```

Figure 87. CBRISQL0 SAMPLIB Member (Part 5 of 5)

CBRISQLX

SAMPLIB member CBRISQLX, as shown in Figure 88 on page 395, provides DB2 definitions for part of the OAM administration database. This job must be modified and run successfully before OAM is used.

```
//CBRISQLX JOB  MSGLEVEL=(1,1),MSGCLASS=A
//***************************************************************
//*
//*  $SEG(CBRISQLX) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Database Definition Job (for Administration
//*  Databases).
//*
//*  This job will create the OAM Administration databases,
//*  tables, and indexes in DB2.
//*
//*  Before running this job, you must change the following:
//*
//*  1. Change "cat_name" to the DB2 VCAT name used
//*     for defining the VSAM data sets in CBRIALCX.
//*
//*  2. Change "auth_id" to the identifier(s)
//*     authorized for the respective group.
//*
//*  3. Change the name in the DSN SYSTEM(DB2) statement to
//*     the name of the DB2 Subsystem in your installation.
//*
//*  4. Change the data set name in the RUN statement
//*     LIB('DB2.RUNLIB.LOAD') phrase to the data set name used
//*     in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  5. Change the PLAN name (DSNTIA21) in the RUN statement to
//*     match your current DB2 version and release level.
//*
//*  6. If you plan to run OAM under DB2 4.1 or above level
//*     and you have specified the Default Index Type = 2 when
//*     you installed the DB2, change "CREATE UNIQUE INDEX" to
//*     "CREATE TYPE 1 UNIQUE INDEX" and change "CREATE INDEX"
//*     to "CREATE TYPE 1 INDEX."
//*
//***************************************************************
//CREATE EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA21) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
          CREATE DATABASE   OAMADMIN;
          COMMIT;
          CREATE TABLESPACE MCIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
          CREATE TABLESPACE SCIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
          CREATE TABLESPACE COLIND
                 IN         OAMADMIN
                 USING      VCAT cat_name
                 LOCKSIZE   ANY
                 CLOSE      NO
                 BUFFERPOOL BP0;
```

Figure 88. CBRISQLX SAMPLIB Member (Part 1 of 3)

```
COMMIT;
CREATE TABLE OAMADMIN.CBR_MGT_CLASS_TBL
       (
        ODMCNUM  SMALLINT    NOT NULL,
        ODMCNAME VARCHAR(30) NOT NULL
       )
       IN OAMADMIN.MCIND;
CREATE UNIQUE INDEX OAMADMIN.CBRMGTX
       ON OAMADMIN.CBR_MGT_CLASS_TBL
       (
        ODMCNUM  ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       SUBPAGES   1
       BUFFERPOOL BP1
       PCTFREE    10;
COMMIT;
CREATE TABLE OAMADMIN.CBR_STO_CLASS_TBL
       (
        ODSCNUM  SMALLINT    NOT NULL,
        ODSCNAME VARCHAR(30) NOT NULL
       )
       IN OAMADMIN.SCIND;
CREATE UNIQUE INDEX OAMADMIN.CBRSTOX
       ON OAMADMIN.CBR_STO_CLASS_TBL
       (
        ODSCNUM  ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       SUBPAGES   1
       BUFFERPOOL BP1
       PCTFREE    10;
COMMIT;
CREATE TABLE OAMADMIN.CBR_COLLECTION_TBL
       (
        ODCLSCNM VARCHAR(30) NOT NULL,
        ODCLMCNM VARCHAR(30) NOT NULL,
        ODCLSGNM VARCHAR(30) NOT NULL,
        ODCLID   INTEGER     NOT NULL,
        ODCLNAME VARCHAR(44) NOT NULL
       )
       IN OAMADMIN.COLIND;
CREATE UNIQUE INDEX OAMADMIN.CBRCLTX1
       ON OAMADMIN.CBR_COLLECTION_TBL
       (
        ODCLID   ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       SUBPAGES   1
       BUFFERPOOL BP1
       PCTFREE    10;
CREATE UNIQUE INDEX OAMADMIN.CBRCLTX2
       ON OAMADMIN.CBR_COLLECTION_TBL
       (
        ODCLNAME ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       SUBPAGES   1
       BUFFERPOOL BP1
       PCTFREE    10;
```

Figure 88. CBRISQLX SAMPLIB Member (Part 2 of 3)

```
CREATE INDEX OAMADMIN.CBRCLTX3
       ON OAMADMIN.CBR_COLLECTION_TBL
       (
        ODCLSGNM ASC
       )
       USING      VCAT cat_name
       CLOSE      NO
       SUBPAGES   1
       BUFFERPOOL BP1
       PCTFREE    10;
COMMIT;
GRANT ALL ON
       OAMADMIN.CBR_MGT_CLASS_TBL
     TO auth_id;
GRANT ALL ON
       OAMADMIN.CBR_STO_CLASS_TBL
     TO auth_id;
GRANT ALL ON
       OAMADMIN.CBR_COLLECTION_TBL
     TO auth_id;
COMMIT;
```

Figure 88. CBRISQLX SAMPLIB Member (Part 3 of 3)

CBRISQLY

SAMPLIB member CBRISQLY, as shown in Figure 89 on page 398, provides the DB2 definitions for part of the OAM administration database. This job must be modified and run successfully before OAM is used.

```
//CBRISQLY JOB  MSGLEVEL=(1,1),MSGCLASS=A
//***********************************************************
//*
//*  $SEG(CBRISQLY) COMP(OSR) PROD(OAM):
//*
//*  OAM DB2 Database Definition Job (for Administration
//*  Databases).
//*
//*  This job will create additional unique indexes for
//*  the OAM Administration Databases in DB2.
//*
//*  Prior to executing this job you need to make the
//*  following modifications:
//*
//*  1. Change "cat_name" to the DB2 VCAT name used
//*     for defining the VSAM data sets in CBRIALCY.
//*
//*  2. Change the name in the DSN SYSTEM(DB2) statement to
//*     the name of the DB2 Subsystem in your installation.
//*
//*  3. Change the data set name in the RUN statement
//*     LIB('DB2.RUNLIB.LOAD') phrase to the data set name used
//*     in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  4. Change the PLAN name (DSNTIA21) in the RUN statement to
//*     match your current DB2 version and release level.
//*
//*  5. If you plan to run OAM under DB2 4.1 or above level
//*     and you have specified the Default Index Type = 2 when
//*     you installed the DB2, change "CREATE UNIQUE INDEX" to
//*     "CREATE TYPE 1 UNIQUE INDEX".
//*
//****************************************************************
//CREATE EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA21) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
          CREATE UNIQUE INDEX OAMADMIN.CBRMGTY
                 ON OAMADMIN.CBR_MGT_CLASS_TBL
                 (
                  ODMCNAME ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 SUBPAGES   1
                 BUFFERPOOL BP1
                 PCTFREE    10;
          CREATE UNIQUE INDEX OAMADMIN.CBRSTOY
                 ON OAMADMIN.CBR_STO_CLASS_TBL
                 (
                  ODSCNAME ASC
                 )
                 USING      VCAT cat_name
                 CLOSE      NO
                 SUBPAGES   1
                 BUFFERPOOL BP1
                 PCTFREE    10;
          COMMIT;
```

Figure 89. CBRISQLY SAMPLIB Member

OAM Optical Configuration Database

SAMPLIB members CBRSAMPL, CBRSMB2, CBRSM150, CBRSMR13, CBRSMERG, and CBRSG100 help you install or migrate to the current release of OAM:

- **CBRSAMPL** is used to install the Optical Configuration Database in a new installation. For an example of this SAMPLIB member, see "CBRSAMPL".
- **CBRSMB2** performs the migration necessary for the TAPEVOL table to add a column used for the support of the IBM 3590-E1x tape device. For an example of this SAMPLIB member, see "CBRSMB2" on page 407.
- **CBRSM150** performs a migration job that adds additional columns to the LIBRARY, DRIVE, VOLUME, and TAPEVOL Optical Configuration Database tables. For an example of this SAMPLIB member, see "CBRSM150" on page 409.
- **CBRSMR13** performs the migration necessary to support the multiple object backup storage groups function and the second backup copies of objects function in OAM/OSMC. It adds a new column BKTYPE to the existing TAPEVOL table. It also performs the migration from the base version of the OAM Object directory tables to the current version, which supports second backup copies of objects. It adds new columns ODBK2LOC and ODBK2SEC to the existing Object directory tables. For an example of this SAMPLIB member, see "CBRSMR13" on page 411.
- **CBRSMERG** performs a database merge of two OAM configuration databases, while **CBRSG100** performs a database merge of two OAM object storage and administration databases. Both of these jobs are executed to allow DB2 data sharing in an OAMplex. For examples of these SAMPLIB members, see "CBRSMERG" on page 415 and "CBRSG100" on page 422.

CBRSAMPL

SAMPLIB member CBRSAMPL, as shown in Figure 90 on page 400, creates the optical configuration database.

```
//CBRSAMPL JOB MSGLEVEL=(1,1),MSGCLASS=A
//*******************************************************************
//*
//*  $SEG(CBRSAMPL) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Definition Job (for Optical Configuration
//*  Database).
//*
//*  This job will create the LCS databases, tables, and indexes
//*  in DB2 for the Optical Configuration Database.
//*
//*  Before running this job, you must change the following:
//*
//*  1. Change "vol_ser" to the volume serials that your
//*     target database should reside on.
//*
//*  2. Change "cat_name" to the name of the catalog you
//*     will be using under DB2.
//*
//*  3. Change "pass_word" to the name of the catalog
//*     password.
//*
//*  4. Change the name in the DSN SYSTEM(DB2) statement to
//*     the name of the DB2 Subsystem in your installation.
//*
//*  5. Change the data set name in the RUN statement
//*     LIB('DBxxx.RUNLIB.LOAD') phrase to the data set name used
//*     in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  6. Change the PLAN name (DSNTIAxx) in the RUN statement to
//*     match your current DB2 version and release level.
//*
//*  CHANGE ACTIVITY:
//*    $L0=JDP3227 320 890601 TUCJRL: Initial Release
//*    $L1=JDP3331 331 900815 TUCJRL: Added the deleted objects
//*                                   table. Changed NUMSLOTS and
//*                                   NUMESLOT to INTEGER in
//*                                   OLIBRARY table. Added DEVTYPE
//*                                   and LIBRDES to OLIBRARY table.
//*                                   Added DEVTYPE and DRIVDES to
//*                                   DRIVE table. Added FRESPACE,
//*                                   DELSPACE, DELCOUNT, FRAGIDX,
//*                                   MEDIATYP, CREDATE, ERRSTAT,
//*                                   VOLEMPTY, and RECOUNT to
//*                                   VOLUME table.
//*    $P1=KBI0238 331 900904 TUCKHB: CHANGED LABEL FOR VOLUMSET
//*                                   TO STORAGE_GROUP.
//*
//*    $L2=CAPELLA2 120 921203 TUCGRD: ADDED CLIBRARY, MEDIATYP AND
//*                                   LIBINDEX TO OLIBRARY TABLE.
//*                                   ADDED CAPACITY TO VOLUME TABLE
//*
//*    $L3=FRBTAPE  332 920220 TUCKHB: ADDED TAPEVOL TABLE FOR
//*                                   FRB TAPE SUPPORT.
//*
//*    $L4=OBJTAPE  120 930523 TUCKMF: UPDATED TAPEVOL STEPS TO ADD
//*                                   NEW 120 COLUMNS FOR LOGICAL
//*                                   KB DELETED AND TAPE COMPACTION
//*                                   INDICATOR.
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 1 of 8)

```
//*    $L5=OPTSPE   130 950823 TUCSMC: Added new DRIVENUM to the
//*                                    DRIVE table.
//*    $L6=HDZ11D0  140 970331 TUCSPP: Specify TYPE 1 INDEX for
//*                                    DB2 4.1 or above level
//*    $L7=HDZ11E0  150 970812 TUCLJT: ADDED MEMBER TO OLIBRARY TABLE,
//*                                    DRIVE TABLE, VOLUME TABLE, AND
//*                                    TAPEVOL TABLE
//*                                    ADDED PLIBRARY TO OLIBRARY TABLE
//*                                    AND VOLUME TABLE
//*    $L8=OW38975  1E0 990609 TUCSPP: ADDED NEW EPI FIELD TO THE
//*                                    TAPE VOLUME TABLE
//*    $L9=HDZ11G0  R13 000915 TUCLJT: ADDED BKTYPE TO VOLUME TABLE AND
//*                                    TAPEVOL TABLE
//*
//*********************************************************************
//OCDBTABS EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
      LIB('DBxxx.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  CREATE STOGROUP CBROAM
      VOLUMES (vol_ser)
      VCAT cat_name
      PASSWORD pass_word;

  CREATE DATABASE CBROAM
      STOGROUP CBROAM
      BUFFERPOOL BP0;

  CREATE TABLESPACE OCLIBTSP
     IN CBROAM
     USING STOGROUP CBROAM
     BUFFERPOOL BP0
     CLOSE NO;

  CREATE TABLESPACE OCDRVTSP
     IN CBROAM
     USING STOGROUP CBROAM
     BUFFERPOOL BP0
     CLOSE NO;

  CREATE TABLESPACE OCSLTTSP
     IN CBROAM
     USING STOGROUP CBROAM
     BUFFERPOOL BP0
     CLOSE NO;

  CREATE TABLESPACE OCVOLTSP
     IN CBROAM
     USING STOGROUP CBROAM
     BUFFERPOOL BP0
     CLOSE NO;
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 2 of 8)

```
CREATE TABLESPACE OCDELTSP
   IN CBROAM
   USING STOGROUP CBROAM
   BUFFERPOOL BP0
   CLOSE NO;

CREATE TABLESPACE OCTVLTSP
   IN CBROAM
   USING STOGROUP CBROAM
   BUFFERPOOL BP0
   CLOSE NO;

CREATE TABLE OLIBRARY
  (NAME      CHAR(8)    NOT NULL,
   ONLINE    CHAR(1)    NOT NULL,
   OPERATNL  CHAR(1)    NOT NULL,
   PATHSTAT  CHAR(1)    NOT NULL,
   COMMAND   CHAR(5)    NOT NULL WITH DEFAULT,
   PRIMCTC   CHAR(4)    NOT NULL WITH DEFAULT,
   PRIMPORT  CHAR(1)    NOT NULL WITH DEFAULT,
   ALTCTC    CHAR(4)    NOT NULL WITH DEFAULT,
   ALTPORT   CHAR(1)    NOT NULL WITH DEFAULT,
   FAULT     CHAR(3)    NOT NULL WITH DEFAULT,
   OLIBTYPE  CHAR(1)    NOT NULL,
   NUMSLOTS  INTEGER    NOT NULL WITH DEFAULT,
   NUMESLOT  INTEGER    NOT NULL WITH DEFAULT,
   NUMDRVS   SMALLINT   NOT NULL,
   RCOMMAND  CHAR(5)    NOT NULL WITH DEFAULT,
   DEVTYPE   CHAR(8)    NOT NULL WITH DEFAULT,
   LIBRDES   CHAR(120)  NOT NULL WITH DEFAULT,
   CLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT,
   MEDIATYP  CHAR(8)    NOT NULL WITH DEFAULT,
   LIBINDEX  SMALLINT   NOT NULL WITH DEFAULT,
   MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,
   PLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT)
  IN CBROAM.OCLIBTSP;

CREATE TABLE DRIVE
  (NAME      CHAR(8)    NOT NULL,
   OLIBRARY  CHAR(8)    NOT NULL,
   CTC       CHAR(4)    NOT NULL,
   SCSI      CHAR(1)    NOT NULL,
   LUN       CHAR(1)    NOT NULL,
   ONLINE    CHAR(1)    NOT NULL,
   OPERATNL  CHAR(1)    NOT NULL,
   LDRIVENO  CHAR(1)    NOT NULL WITH DEFAULT,
   DRIVTYPE  CHAR(1)    NOT NULL,
   DEVTYPE   CHAR(8)    NOT NULL WITH DEFAULT,
   DRIVDES   CHAR(120)  NOT NULL WITH DEFAULT,
   DRIVENUM  SMALLINT   NOT NULL WITH DEFAULT,
   MEMBER    CHAR(16)   NOT NULL WITH DEFAULT)
  IN CBROAM.OCDRVTSP;

CREATE TABLE SLOT
  (NAME      CHAR(3)    NOT NULL,
   OLIBRARY  CHAR(8)    NOT NULL,
   OCCUPIED  CHAR(1)    NOT NULL,
   OPERATNL  CHAR(1)    NOT NULL,
   VOLSER0   CHAR(6)    NOT NULL WITH DEFAULT,
   VOLSER1   CHAR(6)    NOT NULL WITH DEFAULT)
  IN CBROAM.OCSLTTSP;
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 3 of 8)

```
CREATE TABLE VOLUME
  (VOLSER    CHAR(6)    NOT NULL,
   OVOLSER   CHAR(6)    NOT NULL,
   LOCATION  CHAR(1)    NOT NULL,
   SLOT      CHAR(3)    NOT NULL,
   OLIBRARY  CHAR(8)    NOT NULL,
   SHELFLOC  CHAR(32)   NOT NULL WITH DEFAULT,
   MNTDATE   DATE       NOT NULL WITH DEFAULT,
   WRTDATE   DATE       NOT NULL WITH DEFAULT,
   EXPDATE   DATE       NOT NULL WITH DEFAULT,
   EJECTDAT  DATE       NOT NULL WITH DEFAULT,
   LASTDATA  INTEGER    NOT NULL,
   LASTVTCL  INTEGER    NOT NULL,
   LASTVTCP  INTEGER    NOT NULL,
   VOLUMSET  CHAR(8)    NOT NULL,
   TYPE      CHAR(1)    NOT NULL,
   ORIENT    CHAR(1)    NOT NULL,
   FULL      CHAR(1)    NOT NULL,
   READABLE  CHAR(1)    NOT NULL,
   WRITABLE  CHAR(1)    NOT NULL,
   WRTPROT   CHAR(1)    NOT NULL,
   OWNERP    CHAR(1)    NOT NULL WITH DEFAULT,
   OWNER     CHAR(32)   NOT NULL WITH DEFAULT,
   FRESPACE  INTEGER    NOT NULL WITH DEFAULT,
   DELSPACE  INTEGER    NOT NULL WITH DEFAULT,
   DELCOUNT  INTEGER    NOT NULL WITH DEFAULT,
   FRAGIDX   SMALLINT   NOT NULL WITH DEFAULT,
   MEDIATYP  CHAR(2)    NOT NULL WITH DEFAULT,
   CREDATE   DATE       NOT NULL WITH DEFAULT,
   ERRSTAT   SMALLINT   NOT NULL WITH DEFAULT,
   VOLEMPTY  CHAR(1)    NOT NULL WITH DEFAULT,
   RECOUNT   SMALLINT   NOT NULL WITH DEFAULT,
   CAPACITY  INTEGER    NOT NULL WITH DEFAULT,
   MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,
   PLIBRARY  CHAR(8)    NOT NULL WITH DEFAULT,
   BKTYPE    CHAR(1)    NOT NULL WITH DEFAULT)
  IN CBROAM.OCVOLTSP;

CREATE TABLE DELOBJT
  (COLNAME   CHAR(44)   NOT NULL,
   OBJNAME   CHAR(44)   NOT NULL,
   VOLSER    CHAR(6)    NOT NULL,
   VTOCTOKN  INTEGER    NOT NULL,
   OBJSIZE   INTEGER    NOT NULL)
  IN CBROAM.OCDELTSP;
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 4 of 8)

```
 CREATE TABLE TAPEVOL
   (VOLSER    CHAR(6)    NOT NULL,
    UNITNAME  CHAR(8)    NOT NULL,
    MEDIATYP  CHAR(2)    NOT NULL,
    STORGRP   CHAR(8)    NOT NULL,
    TYPE      CHAR(1)    NOT NULL,
    CREDATE   DATE       NOT NULL,
    MNTDATE   DATE       NOT NULL,
    WRTDATE   DATE       NOT NULL,
    EXPDATE   DATE       NOT NULL,
    CAPACITY  INTEGER    NOT NULL,
    FRESPACE  INTEGER    NOT NULL,
    LSTBLKID  INTEGER    NOT NULL,
    PFULL     SMALLINT   NOT NULL,
    NUMLBLKS  INTEGER    NOT NULL,
    NUMLKBW   INTEGER    NOT NULL,
    NUMPKBW   INTEGER    NOT NULL,
    NUMLKBDE  INTEGER    NOT NULL,
    FULL      CHAR(1)    NOT NULL,
    READABLE  CHAR(1)    NOT NULL,
    WRITABLE  CHAR(1)    NOT NULL,
    INUSE     CHAR(1)    NOT NULL,
    COPIED    CHAR(1)    NOT NULL,
    AVOLSER   CHAR(6)    NOT NULL,
    COMPACT   CHAR(1)    NOT NULL,
    EPI       SMALLINT   NOT NULL WITH DEFAULT,
    MEMBER    CHAR(16)   NOT NULL WITH DEFAULT,
    BKTYPE    CHAR(1)    NOT NULL WITH DEFAULT)
   IN CBROAM.OCTVLTSP;

CREATE UNIQUE INDEX LNAMINDX
  ON OLIBRARY
  (NAME ASC)
  USING STOGROUP CBROAM
  CLUSTER
  BUFFERPOOL BP0
  CLOSE NO;

CREATE UNIQUE INDEX DNAMINDX
  ON DRIVE
  (NAME ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLOSE NO;

CREATE UNIQUE INDEX DRIDINDX
  ON DRIVE
  (CTC, SCSI, LUN ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

CREATE UNIQUE INDEX SLIBINDX
  ON SLOT
  (NAME, OLIBRARY ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 5 of 8)

```
CREATE UNIQUE INDEX VSERINDX
  ON VOLUME
  (VOLSER ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

CREATE INDEX DVOLINDX
  ON DELOBJT
  (VOLSER ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLOSE NO;

CREATE UNIQUE INDEX DELOINDX
  ON DELOBJT
  (COLNAME, OBJNAME, VOLSER, VTOCTOKN ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

CREATE UNIQUE INDEX TVOLINDX
  ON TAPEVOL
  (VOLSER ASC)
  USING STOGROUP CBROAM
  BUFFERPOOL BP0
  CLUSTER
  CLOSE NO;

LABEL ON OLIBRARY
  (NAME IS 'NAME',
   ONLINE IS 'ONLINE',
   OPERATNL IS 'OPERATIONAL',
   PATHSTAT IS 'CURRENT_PATH',
   COMMAND IS 'CURRENT_COMMAND',
   PRIMCTC IS 'PRIMARY_CTC',
   PRIMPORT IS 'PRIMARY_PORT',
   ALTCTC IS 'ALTERNATE_CTC',
   ALTPORT IS 'ALTERNATE_PORT',
   FAULT IS 'FAULT_CODE',
   OLIBTYPE IS 'LIBRARY_TYPE',
   NUMSLOTS IS 'SLOTS',
   NUMESLOT IS 'EMPTY_SLOTS',
   NUMDRVS IS 'DRIVES',
   RCOMMAND IS 'RECOVERY_COMMAND',
   DEVTYPE IS 'DEVICE_TYPE',
   LIBRDES IS 'LIBRARY_DESCRIPTION',
   CLIBRARY IS 'CONTROLLING_LIBRARY',
   MEDIATYP IS 'DEFAULT_MEDIA_TYPE',
   LIBINDEX IS 'LIBRARY_INDEX',
   MEMBER IS 'OAM_XCF_MEMBER',
   PLIBRARY IS 'DEFAULT_PSEUDO_LIBRARY');
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 6 of 8)

```
LABEL ON DRIVE
  (NAME IS 'NAME',
   OLIBRARY IS 'OLIBRARY',
   CTC IS 'CTC',
   SCSI IS 'SCSI',
   LUN IS 'LUN',
   ONLINE IS 'ONLINE',
   OPERATNL IS 'OPERATIONAL',
   LDRIVENO IS 'DRIVE_NUMBER',
   DRIVTYPE IS 'DRIVE_TYPE',
   DEVTYPE IS 'DEVICE_TYPE',
   DRIVDES IS 'DRIVE_DESCRIPTION',
   DRIVENUM IS 'PHYS_DRIVE_NUMBER',
   MEMBER IS 'OAM_XCF_MEMBER');

LABEL ON SLOT
  (NAME IS 'NAME',
   OLIBRARY IS 'OLIBRARY',
   OCCUPIED IS 'OCCUPIED',
   OPERATNL IS 'OPERATIONAL',
   VOLSER0 IS 'VOLSER0',
   VOLSER1 IS 'VOLSER1');

LABEL ON VOLUME
  (VOLSER IS 'VOLSER',
   OVOLSER IS 'OTHER_VOLSER',
   LOCATION IS 'LOCATION',
   SLOT IS 'SLOT',
   OLIBRARY IS 'OLIBRARY',
   SHELFLOC IS 'SHELF_LOCATION',
   MNTDATE IS 'DATE_LAST_MOUNTED',
   WRTDATE IS 'DATE_LAST_WRITTEN',
   EXPDATE IS 'EXPIRATION_DATE',
   EJECTDAT IS 'EJECT/ENTER_DATE',
   LASTDATA IS 'LAST_DATA_SECTOR',
   LASTVTCL IS 'LAST_LOGICAL_VTOC_SECTOR',
   LASTVTCP IS 'LAST_PHYSICAL_VTOC_SECTOR',
   VOLUMSET IS 'STORAGE_GROUP',
   TYPE IS 'TYPE',
   ORIENT IS 'ORIENTATION',
   FULL IS 'FULL',
   READABLE IS 'VOLUME_READABLE_STATUS',
   WRITABLE IS 'VOLUME_WRITABLE_STATUS',
   WRTPROT IS 'WRITE_PROTECTED',
   OWNERP IS 'OWNER_INFORMATION_POSITION',
   OWNER IS 'OWNER_INFORMATION',
   FRESPACE IS 'FREE_SPACE',
   DELSPACE IS 'DELETED_SPACE',
   DELCOUNT IS 'DELETED_OBJECTS',
   FRAGIDX IS 'FRAGMENTATION_INDEX',
   MEDIATYP IS 'MEDIA_TYPE',
   CREDATE IS 'CREATE_DATE',
   ERRSTAT IS 'VOLUME_ERROR_STATUS',
   VOLEMPTY IS 'VOLUME_EMPTY',
   RECOUNT IS 'DELETED_OBJECTS_RECOUNT',
   CAPACITY IS 'CAPACITY',
   MEMBER IS 'OAM_XCF_MEMBER',
   PLIBRARY IS 'PSEUDO_LIBRARY_FOR_VOLUME',
   BKTYPE IS 'BACKUP_TYPE');
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 7 of 8)

```
 LABEL ON DELOBJT
   (COLNAME IS 'COLLECTION_NAME',
    OBJNAME IS 'OBJECT_NAME',
    VOLSER IS 'VOLSER',
    VTOCTOKN IS 'VTOC_TOKEN',
    OBJSIZE IS 'OBJECT_SIZE');

 LABEL ON TAPEVOL
   (VOLSER IS 'VOLSER',
    UNITNAME IS 'UNIT_NAME',
    MEDIATYP IS 'MEDIA_TYPE',
    STORGRP IS 'STORAGE_GROUP',
    TYPE IS 'TYPE',
    CREDATE IS 'CREATION_DATE',
    MNTDATE IS 'DATE_LAST_MOUNTED',
    WRTDATE IS 'DATE_LAST_WRITTEN',
    EXPDATE IS 'EXPIRATION_DATE',
    CAPACITY IS 'CAPACITY',
    FRESPACE IS 'FREE_SPACE',
    LSTBLKID IS 'LAST_BLOCKID',
    PFULL IS 'PERCENT_FULL',
    NUMLBLKS IS 'LOGICAL_BLOCKS_WRITTEN',
    NUMLKBW IS 'LOGICAL_KILOBYTES_WRITTEN',
    NUMPKBW IS 'PHYSICAL_KILOBYTES_WRITTEN',
    NUMLKBDE IS 'LOGICAL_KILOBYTES_DELETED',
    FULL IS 'FULL',
    READABLE IS 'VOLUME_READABLE_STATUS',
    WRITABLE IS 'VOLUME_WRITABLE_STATUS',
    INUSE IS 'IN_USE',
    COPIED IS 'COPIED',
    AVOLSER IS 'ALTERNATE_VOLUME',
    COMPACT IS 'TAPE_COMPACTION_INDICATOR',
    EPI     IS 'EPI',
    MEMBER IS 'OAM_XCF_MEMBER',
    BKTYPE IS 'BACKUP_TYPE');


 COMMIT;
/*
```

Figure 90. CBRSAMPL SAMPLIB Member (Part 8 of 8)

CBRSMB2

SAMPLIB member CBRSMB2, as shown in Figure 91 on page 408, performs the migration necessary for the TAPEVOL table to add a column used for the support of the IBM 3590-E1x tape device.

```
//CBRSMB2 JOB MSGLEVEL=(1,1),MSGCLASS=A
//*******************************************************************
//*
//*  $SEG(CBRSMB2) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Migration Jobs (for Optical Configuration
//*  Database TAPEVOL Table).
//*
//*  This job performs the migration of the TAPEVOL
//*  table to the new version of the TAPEVOL table for
//*  IBM 3590-E1x tape device support.
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing TAPEVOL table prior to running the CBRSMB2  job.
//*
//*  ---------------------------------------------------------------
//*
//*  Before running these jobs, you must change the following:
//*
//*  1. Change the name in the DSN SYSTEM(DB2) statement to
//*     the name of the DB2 Subsystem in your installation.
//*
//*  2. Change the data set name in the RUN statement
//*     LIB('DB2.RUNLIB.LOAD') phrase to the data set name used
//*     in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  3. Change the PLAN name (DSNTIAxx) in the RUN statement to
//*     match your current DB2 version and release level.
//*
//*  ---------------------------------------------------------------
//*
//*
//*  STEP        DESCRIPTION
//*  --------    -----------
//*
//*  ALTERTVL -  This step adds the new EPI column
//*              to the TAPEVOL table.
//*
//*  LABELTVL -  This step provides the new label for the EPI column.
//*
//*  CHANGTVL - This step sets the EPI field to 255 ('FF'X)
//*             for each tape volume in the TAPEVOL table
//*             which means the volume was added to the
//*             OAM Tape Volume Table before the
//*             IBM 3590-E1X tape device support.
//*    NOTE:
//*    A return code of 4 is expected from CHANGTVL step
//*    because SQL issues a warning for an  UPDATE statement
//*    that does not include a WHERE clause.
//*    (SQLSTATE = 01504)
//*
//*  ---------------------------------------------------------------
//*
//*
//*   CHANGE ACTIVITY:
//*    $L0=OW38975  1E0 990611 TUCSPP: 3590-E1X TAPE DEVICE SUPPORT
//*
```

Figure 91. CBRSMB2 SAMPLIB Member (Part 1 of 2)

```
//*********************************************************************
//*  Alter the TAPEVOL table to add the new EPI
//*  column  for 3590-E1X tape device support.
//*********************************************************************
//ALTERTVL EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 ALTER TABLE TAPEVOL ADD EPI SMALLINT      NOT NULL WITH DEFAULT;

 COMMIT;
/*
//*********************************************************************
//*  Place label on TAPEVOL table EPI column
//*********************************************************************
//LABELTVL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LABEL ON TAPEVOL
   (EPI      IS 'EPI');

 COMMIT;
/*
//*********************************************************************
//*  Update the TAPEVOL table to set EPI value to 255 ('FF'X)
//*  which means that the volume was added to the TAPEVOL table
//*  before the 3590-E1X tape device support.
//*
//*  Note: Since EPI is defined not null with default, it will
//*        be set to 0 for all existing rows in the TAPEVOL table.
//*        EPI = '00'X means that the volume is first written by
//*                    a native tape device (NOT in emulation mode).
//*********************************************************************
//CHANGTVL EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIAxx) -
      LIB('DB2.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 UPDATE TAPEVOL
   SET EPI = 255;

 COMMIT;

/*
```

Figure 91. CBRSMB2 SAMPLIB Member (Part 2 of 2)

CBRSM150

SAMPLIB member CBRSM150, as shown in Figure 92 on page 410, migrates the Optical Configuration Database from a DFSMS 1.4.0 release to the DFSMS 1.5.0 release to provide OAM support for the Parallel Sysplex capability.

```
//CBRSM150 JOB MSGLEVEL=(1,1),MSGCLASS=A
//*********************************************************************
//*
//*  $SEG(CBRSM150) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Migration Job (for the Optical Configuration
//*  Database).
//*
//*  ---------------------------------------------------------------
//*  ---------------------------------------------------------------
//*
//*  CBRSM150
//*
//*  This job will perform the migration from the DFSMS/MVS 1.4.0
//*  (or DFSMS/MVS 1.3.0 after running CBRSM131 conversion job)
//*  version of the Optical Configuration Database to the DFSMS/MVS
//*  1.5.0 version which supports OAM in a sysplex environment
//*  using XCF communications.
//*
//*  This job will:
//*   1. add a new column MEMBER to the existing OLIBRARY, DRIVE,
//*      VOLUME, and TAPEVOL tables.
//*   2. add a new column PLIBRARY to the existing OLIBRARY table
//*   3. add a new column PLIBRARY to the existing VOLUME table and
//*      prime it with the value in the LIBRARY table, if the volume
//*      record currently indicates that the volume is shelf resident
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing OLIBRARY, DRIVE, VOLUME, and TAPEVOL tables prior
//*  to executing this CBRSM150 job.
//*
//*  Before running this job, you must change the following:
//*
//*    1.Change the name in the DSN SYSTEM(DB2) statement to
//*      the name of the DB2 Subsystem in your installation.
//*
//*    2.Change the PLAN name (DSNTIA31) in the RUN statement to
//*      match your current DB2 version and release level.
//*
//*    3.Change the data set name in the RUN statement
//*      LIB('DB310.RUNLIB.LOAD') phrase to the data set name used
//*      in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  CHANGE ACTIVITY:
//*    $L0=HDZ11E0  150 970812 TUCLJT: Initial Release
//*
//*********************************************************************
//*********************************************************************
//*  Alter the OLIBRARY table to add the MEMBER and PLIBRARY
//*  column definitions.
//*  Alter the DRIVE table to add the MEMBER column definition
//*  Alter the VOLUME table to add the MEMBER and PLIBRARY
//*  column definitions.
//*  Alter the TAPEVOL table to add the MEMBER column definition
//*********************************************************************
//ALTERTAB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
```

Figure 92. CBRSM150 SAMPLIB Member (Part 1 of 2)

```
 ALTER TABLE OLIBRARY ADD MEMBER CHAR(16) NOT NULL WITH DEFAULT;
 ALTER TABLE OLIBRARY ADD PLIBRARY CHAR(8) NOT NULL WITH DEFAULT;

 ALTER TABLE DRIVE ADD MEMBER CHAR(16) NOT NULL WITH DEFAULT;

 ALTER TABLE VOLUME ADD MEMBER CHAR(16) NOT NULL WITH DEFAULT;
 ALTER TABLE VOLUME ADD PLIBRARY CHAR(8) NOT NULL WITH DEFAULT;

 ALTER TABLE TAPEVOL ADD MEMBER CHAR(16) NOT NULL WITH DEFAULT;

 COMMIT;
/*
//*********************************************************************
//*  Place label in tables for new columns
//*********************************************************************
//LABELTAB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LABEL ON OLIBRARY
   (MEMBER IS 'OAM_XCF_MEMBER',
    PLIBRARY IS 'DEFAULT_PSEUDO_LIBRARY');

 LABEL ON DRIVE
   (MEMBER IS 'OAM_XCF_MEMBER');

 LABEL ON VOLUME
   (MEMBER IS 'OAM_XCF_MEMBER',
    PLIBRARY IS 'PSEUDO_LIBRARY_FOR_VOLUME');

 LABEL ON TAPEVOL
   (MEMBER IS 'OAM_XCF_MEMBER');

 COMMIT;
/*
//*********************************************************************
//*  Prime PLIBRARY field with value from OLIBRARY field for
//*  shelf resident volumes
//*********************************************************************
//PRIMVPLB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 UPDATE VOLUME SET PLIBRARY = OLIBRARY WHERE LOCATION = 'S';

 COMMIT;
/*
```

Figure 92. CBRSM150 SAMPLIB Member (Part 2 of 2)

CBRSMR13

SAMPLIB member CBRSMR13, as shown in Figure 93 on page 412, performs the migration necessary to support the multiple object backup storage groups function and the second backup copies of objects function in OAM/OSMC. It also performs the migration from the base version of the OAM Object directory tables to the current version, which supports second backup copies of objects. This PARMLIB member has two jobs. SMR13A adds a new column BKTYPE to the existing to the existing TAPEVOL table and a new column BKTYPE to the existing VOLUME table. SMR13B adds new columns ODBK2LOC and ODBK2SEC to the existing Object directory tables.

```
//SMR13A   JOB MSGLEVEL=(1,1),MSGCLASS=A
//*********************************************************************
//*
//*  $SEG(SMR13A) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Migration Job (for the Optical Configuration
//*  Database).
//*
//*  ---------------------------------------------------------------
//*  ---------------------------------------------------------------
//*
//*  SMR13A
//*
//*  This job will perform the migration from the DFSMS/MVS 1.5.0
//*  or OS/390 R10 version of the Optical Configuration Database
//*  to the z/OS V1 R3 version which supports the multiple object
//*  backup storage groups and the second object backup function
//*  in OAM/OSMC.
//*
//*  This job will:
//*   1. add a new column BKTYPE to the existing VOLUME table
//*   2. add a new column BKTYPE to the existing TAPEVOL table
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing VOLUME and TAPEVOL tables prior to executing this
//*  migration job for recovery purposes.
//*
//*  Before running this job, you must change the following:
//*
//*    1.Change the name in the DSN SYSTEM(DB2) statement to
//*      the name of the DB2 Subsystem in your installation.
//*
//*    2.Change the PLAN name (DSNTIA31) in the RUN statement to
//*      match your current DB2 version and release level.
//*
//*    3.Change the data set name in the RUN statement
//*      LIB('DB310.RUNLIB.LOAD') phrase to the data set name used
//*      in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*  CHANGE ACTIVITY:
//*    $L0=HDZ11G0  R13 000917 TUCLJT: Initial Release
//*
//*********************************************************************
```

Figure 93. CBRSMR13 SAMPLIB Member (Part 1 of 4)

```
//**********************************************************************
//*  Alter the VOLUME table to add the BKTYPE column definition
//*  Alter the TAPEVOL table to add the BKTYPE column definition
//**********************************************************************
//ALTERTAB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 ALTER TABLE VOLUME ADD BKTYPE CHAR(1) NOT NULL WITH DEFAULT;

 ALTER TABLE TAPEVOL ADD BKTYPE CHAR(1) NOT NULL WITH DEFAULT;

 COMMIT;
/*
//**********************************************************************
//*  Place label in tables for new columns
//**********************************************************************
//LABELTAB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LABEL ON VOLUME
   (BKTYPE IS 'BACKUP_TYPE');

 LABEL ON TAPEVOL
   (BKTYPE IS 'BACKUP_TYPE');

 COMMIT;
/*
//**********************************************************************
//*  Prime BKTYPE field with '1' (for backup type 1) for backup volumes
//**********************************************************************
//PRIMVPLB EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 UPDATE VOLUME SET BKTYPE = '1' WHERE TYPE = 'B';

 UPDATE TAPEVOL SET BKTYPE = '1' WHERE TYPE = 'B';

 COMMIT;
/*
```

Figure 93. CBRSMR13 SAMPLIB Member (Part 2 of 4)

```
//SMR13B   JOB MSGLEVEL=(1,1),MSGCLASS=A
//*********************************************************************
//*
//*  $SEG(SMR13B) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Migration Job (for the OAM Object Directories).
//*
//*  ---------------------------------------------------------------
//*  ---------------------------------------------------------------
//*
//*  SMR13B
//*
//*  This job will perform the migration from the base version of
//*  the OAM Object directory tables to the z/OS V1 R3 version,
//*  which supports second backup copies of objects.
//*
//*  This job will:
//*   1. add new columns ODBK2LOC and ODBK2SEC to the existing Object
//*      Directory tables.
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing Object Directory tables prior to executing this
//*  migration job for recovery purposes.
//*
//*  Before running this job, you must change the following:
//*
//*    1.Change the name in the DSN SYSTEM(DB2) statement to
//*      the name of the DB2 Subsystem in your installation.
//*
//*    2.Change the PLAN name (DSNTIA31) in the RUN statement to
//*      match your current DB2 version and release level.
//*
//*    3.Change the data set name in the RUN statement
//*      LIB('DB310.RUNLIB.LOAD') phrase to the data set name used
//*      in your installation for the DB2 RUNLIB.LOAD data set.
//*
//*    4.Update ALTER, DROP, and CREATE statements:
//*
//*     a.Repeat the set of ALTER, CROP, and CREATE statements
//*       for every Object Directory (Object Storage Group)
//*       in your configuration
//*
//*     b.Modify the table high level qualifier (hlq) in each of the
//*       statements to match that of a high level qualifier (for an
//*       Object Storage Group) in your configuratoin.
//*
//*     EXAMPLE:
//*       A configuration consists of 2 object storage groups,
//*       OBJGRP1 and OBJGRP2, with high level qualifiers of
//*       GRP1TBL and GRP2TBL, respectively.  The resulting
//*       statements would be:
//*
//*       ALTER TABLE GRP1TBL.OSM_OBJ_DIR ADD ODBK2LOC CHAR(6)
//*         NOT NULL WITH DEFAULT;
//*       ALTER TABLE GRP1TBL.OSM_OBJ_DIR ADD ODBK2SEC INTEGER
//*         NOT NULL WITH DEFAULT;
//*       DROP VIEW GRP1TBL.V_OSM_OBJ_DIR;
//*       CREATE VIEW GRP1TBL.V_OSM_OBJ_DIR
//*         AS SELECT ALL * FROM GRP1TBL.OSM_OBJ_DIR;
//*
```

Figure 93. CBRSMR13 SAMPLIB Member (Part 3 of 4)

```
//*       ALTER TABLE GRP2TBL.OSM_OBJ_DIR ADD ODBK2LOC CHAR(6)
//*         NOT NULL WITH DEFAULT;
//*       ALTER TABLE GRP2TBL.OSM_OBJ_DIR ADD ODBK2SEC INTEGER
//*         NOT NULL WITH DEFAULT;
//*       DROP VIEW GRP2TBL.V_OSM_OBJ_DIR;
//*       CREATE VIEW GRP2TBL.V_OSM_OBJ_DIR
//*         AS SELECT ALL * FROM GRP2TBL.OSM_OBJ_DIR;
//*
//*  CHANGE ACTIVITY:
//*    $L0=HDZ11G0  R13 000917 TUCLJT: Initial Release
//*
//*********************************************************************
//*********************************************************************
//*  Alter the Object Directories to add ODBK2LOC and ODBK2SEC
//*********************************************************************
//ALTERDIR EXEC PGM=IKJEFT01,DYNAMNBR=20,COND=(4,LT)
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB2)
 RUN  PROGRAM(DSNTIAD) PLAN(DSNTIA31) -
      LIB('DB310.RUNLIB.LOAD')
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 ALTER TABLE hlq.OSM_OBJ_DIR ADD ODBK2LOC CHAR(6) NOT NULL WITH DEFAULT;

 ALTER TABLE hlq.OSM_OBJ_DIR ADD ODBK2SEC INTEGER NOT NULL WITH DEFAULT;

 DROP VIEW hlq.V_OSM_OBJ_DIR;

 CREATE VIEW hlq.V_OSM_OBJ_DIR AS SELECT ALL * FROM hlq.OSM_OBJ_DIR;

 COMMIT;
/*
```

Figure 93. CBRSMR13 SAMPLIB Member (Part 4 of 4)

CBRSMERG

SAMPLIB member CBRSMERG, as shown in Figure 94 on page 416, assists you in merging the Optical Configuration Databases (OCDBs) for use with DB2 data sharing in an OAMplex environment. This may not be the only way of performing this task. Use a method that best suits the requirements for your environment. All warnings, prerequisites, or recommendations apply regardless of the method used to perform this merge. If you choose to use this sample job, it must be modified and run successfully before OAM is used.

```
//CBRSMERG JOB MSGLEVEL=(1,1),MSGCLASS=A
//*********************************************************************
//*
//*  $SEG(CBRSMERG) COMP(DBM) PROD(OAM):
//*
//*  OAM DB2 Database Merge Job (combining optical configuration
//*  databases from multiple systems)
//*
//*  ----------------------------------------------------------------
//*  ----------------------------------------------------------------
//*
//*  CBRSMERG
//*
//*  This job is intended to assist in the task of merging OAM
//*  configuration databases for use with DB2 data sharing in an
//*  OAMplex environment.
//*  DFSMS 1.5.0 provides OAM support in a parallel sysplex
//*  environment using XCF communications and DB2 data sharing.
//*
//*
//*  This job will:
//*   1. Load data from other system (DB2 data from other
//*      system's DB2 tables) into this system's OAM's tables.
//*
//*******************************************************************
//*
//*  Before running this job, you MUST obtain the following:
//*    - load the information from the OAM configuration tables
//*      from the other OAM/system where data is to be copied from
//*      onto a dataset on this system
//*      The simplest way to do this is to do an SQL SELECT * from
//*      the OAM configuration database tables on the other systems
//*      and editing the column headers out of the resulting output
//*      so that just the data from the rows remains.
//*    - Note the beginning and ending columns where the data
//*      resides for each column in the table rows.
//*
//*******************************************************************
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing tables:
//*    OAM configuration database tables:
//*       OLIBRARY
//*       DRIVE
//*       VOLUME
//*       TAPEVOL
//*       DELOBJT
//*       SLOT
//*  These tables will be modified directly by this job, so a
//*  backup copy is necessary in case recovery is needed.
//*
//*******************************************************************
//*
//*  Before running this job, you must change the following:
//*
//*    1.Change the PARM='DB2' in the JOB STEP statements to
//*      the name of the DB2 Subsystem in your installation.
//*
//*    2.Change the data set name SYS1.DB2.V4R1M0.SDSNLOAD
//*      in the STEPLIB statements to the data set name used
//*      for the DB2 SDSNLOAD dataset, if necessary.
//*
```

Figure 94. CBRSMERG SAMPLIB Member (Part 1 of 7)

```
//*    3.Change RESUME YES to RESUME NO if you are loading
//*      into empty tables.
//*
//*    4.Change the data set names in the CBRSMERx steps to
//*      the appropriate data set names for loading the DB2
//*      tables from other systems (this sample job is set up
//*      as though datasets are pre-allocated):
//*
//*      smerge.map      = map dataset for DB2 in the job (reused
//*                        in this job, or can use separate data
//*                        sets for each job step if desired
//*      smerge.err      = error dataset for DB2 in the job (reused
//*                        in this job, or can use separate data
//*                        sets for each job step if desired
//*
//*      input.libtable  = DSN with the OLIBRARY table row values
//*                        from the system to be merged
//*      workdsn.forlib  = work dataset for DB2 in job step
//*      sortdsn.forlib  = sort dataset for DB2 in job step
//*      discard.forlib  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      input.slttable  = DSN with the SLOT table row values
//*                        from the system to be merged
//*      workdsn.forslot = work dataset for DB2 in job step
//*      sortdsn.forslot = sort dataset for DB2 in job step
//*      discard.forslot = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      input.drvtable  = DSN with the DRIVE table row values
//*                        from the system to be merged
//*      workdsn.fordrv  = work dataset for DB2 in job step
//*      sortdsn.fordrv  = sort dataset for DB2 in job step
//*      discard.fordrv  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      input.deltable  = DSN with the DELOBJT table row values
//*                        from the system to be merged
//*      workdsn.fordelo = work dataset for DB2 in job step
//*      sortdsn.fordelo = sort dataset for DB2 in job step
//*      discard.fordelo = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      input.voltable  = DSN with the VOLUME table row values
//*                        from the system to be merged
//*      workdsn.forvol  = work dataset for DB2 in job step
//*      sortdsn.forvol  = sort dataset for DB2 in job step
//*      discard.forvol  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      input.tvoltable = DSN with the TAPEVOL table row values
//*                        from the system to be merged
//*      workdsn.fortvol = work dataset for DB2 in job step
//*      sortdsn.fortvol = sort dataset for DB2 in job step
//*      discard.fortvol = DSN for the output of rows that could
//*                        not be merged from the other system
//*
```

Figure 94. CBRSMERG SAMPLIB Member (Part 2 of 7)

```
//*      **NOTE: For these datasets, use size calculations that
//*              would be needed for your installation, using the
//*              DB2 guidelines in the DB2 Command and Utility
//*              Reference for the LOAD utility.
//*
//*    5.Change the POSITION(xx:yy) in the CBRSMER* job steps to
//*      correlate to the actual beginning, (and ending if needed),
//*      columns where the data for each column resides in the input
//*      datasets (the SYSREC DD statement dataset).
//*
//*    6.The integer fields are set up as EXTERNAL(zz) in the job
//*      steps, be sure that any integer values in the columns that
//*      are not the full length are padded with preceding zeroes
//*      in the input dataset (the SYSREC DD statement dataset).
//*
//*********************************************************************
//*
//*  After running this job, do the following:
//*
//*    1.Check the return codes from the job to verify success
//*      or failure of the data merge.
//*
//*    2.Check the data sets below for any rows that could not
//*      be merged into the configuration database.  The most
//*      likely cause of failure would be duplicate rows.
//*
//*      discard.forlib  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      discard.forslot = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      discard.fordrv  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      discard.fordelo = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      discard.forvol  = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//*      discard.fortvol = DSN for the output of rows that could
//*                        not be merged from the other system
//*
//**********************************************************************
//*  Load configuration tables from different DB2 database
//**********************************************************************
//CBRSMER1 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.libtable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forlib,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forlib,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forlib,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
```

Figure 94. CBRSMERG SAMPLIB Member (Part 3 of 7)

```
 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE OLIBRARY
    (NAME       POSITION(xx)  CHAR(8),
     ONLINE     POSITION(xx)  CHAR(1),
     OPERATNL   POSITION(xx)  CHAR(1),
     PATHSTAT   POSITION(xx)  CHAR(1),
     COMMAND    POSITION(xx)  CHAR(5) DEFAULTIF(COMMAND=''),
     PRIMCTC    POSITION(xx)  CHAR(4) DEFAULTIF(PRIMCTC=''),
     PRIMPORT   POSITION(xx)  CHAR(1) DEFAULTIF(PRIMPORT=''),
     ALTCTC     POSITION(xx)  CHAR(4) DEFAULTIF(ALTCTC=''),
     ALTPORT    POSITION(xx)  CHAR(1) DEFAULTIF(ALTPORT=''),
     FAULT      POSITION(xx)  CHAR(3) DEFAULTIF(FAULT=''),
     OLIBTYPE   POSITION(xx)  CHAR(1),
     NUMSLOTS   POSITION(xx)  INTEGER EXTERNAL(3),
     NUMESLOT   POSITION(xx)  INTEGER EXTERNAL(3),
     NUMDRVS    POSITION(xx)  INTEGER EXTERNAL(3),
     RCOMMAND   POSITION(xx)  CHAR(5) DEFAULTIF(RCOMMAND=''),
     DEVTYPE    POSITION(xx)  CHAR(8) DEFAULTIF(RCOMMAND=''),
     LIBRDES    POSITION(xx:yy)  CHAR DEFAULTIF(LIBRDES=''),
     CLIBRARY   POSITION(xx)  CHAR(8) DEFAULTIF(CLIBRARY=''),
     MEDIATYP   POSITION(xx)  CHAR(8) DEFAULTIF(MEDIATYP=''),
     LIBINDEX   POSITION(xx)  INTEGER EXTERNAL(1),
     PLIBRARY   POSITION(xx)  CHAR(16) DEFAULTIF(MEMBER=''),
     MEMBER     POSITION(xx)  CHAR(8) DEFAULTIF(PLIBRARY=''))
/*
//*********************************************************************
//CBRSMER2 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.slttable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forslot,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forslot,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forslot,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE SLOT
    (NAME       POSITION(xx)  CHAR(3),
     OLIBRARY   POSITION(xx)  CHAR(8),
     OCCUPIED   POSITION(xx)  CHAR(1),
     OPERATNL   POSITION(xx)  CHAR(1),
     VOLSER0    POSITION(xx)  CHAR(6) DEFAULTIF(VOLSER0=''),
     VOLSER1    POSITION(xx)  CHAR(6) DEFAULTIF(VOLSER1=''))
 /*
//*********************************************************************
```

Figure 94. CBRSMERG SAMPLIB Member (Part 4 of 7)

```
//CBRSMER3 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.drvtable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.fordrv,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.fordrv,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.fordrv,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE DRIVE
    (NAME       POSITION(xx)    CHAR(8),
     OLIBRARY   POSITION(xx)    CHAR(8),
     CTC        POSITION(xx)    CHAR(4),
     SCSI       POSITION(xx)    CHAR(1),
     LUN        POSITION(xx)    CHAR(1),
     ONLINE     POSITION(xx)    CHAR(1),
     OPERATNL   POSITION(xx)    CHAR(1),
     LDRIVENO   POSITION(xx)    CHAR(1) DEFAULTIF(LDRIVENO=''),
     DRIVTYPE   POSITION(xx)    CHAR(1),
     DEVTYPE    POSITION(xx)    CHAR(8) DEFAULTIF(DEVTYPE=''),
     DRIVEDES   POSITION(xx:yy) CHAR DEFAULTIF(DRIVDES=''),
     DRIVENUM   POSITION(xx)    INTEGER EXTERNAL(1),
     MEMBER     POSITION(xx)    CHAR(16) DEFAULTIF(MEMBER=''))
 /*
//*********************************************************************
//CBRSMER4 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.voltable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forvol,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
```

Figure 94. CBRSMERG SAMPLIB Member (Part 5 of 7)

```
 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE VOLUME
    (VOLSER     POSITION(xx)  CHAR(6),
     OVOLSER    POSITION(xx)  CHAR(6),
     LOCATION   POSITION(xx)  CHAR(1),
     SLOT       POSITION(xx)  CHAR(3),
     OLIBRARY   POSITION(xx)  CHAR(8),
     SHELFLOC   POSITION(xx)  CHAR(32),
     MNTDATE    POSITION(xx)  DATE EXTERNAL(10),
     WRTDATE    POSITION(xx)  DATE EXTERNAL(10),
     EXPDATE    POSITION(xx)  DATE EXTERNAL(10),
     EJECTDAT   POSITION(xx)  DATE EXTERNAL(10),
     LASTDATA   POSITION(xx)  INTEGER EXTERNAL(1),
     LASTVTCL   POSITION(xx)  INTEGER EXTERNAL(1),
     LASTVTCP   POSITION(xx)  INTEGER EXTERNAL(1),
     VOLUMESET  POSITION(xx)  CHAR(8),
     TYPE       POSITION(xx)  CHAR(1),
     ORIENT     POSITION(xx)  CHAR(1),
     FULL       POSITION(xx)  CHAR(1),
     READABLE   POSITION(xx)  CHAR(1),
     WRITABLE   POSITION(xx)  CHAR(1),
     WRTPROT    POSITION(xx)  CHAR(1),
     OWNERP     POSITION(xx)  CHAR(1),
     OWNER      POSITION(xx)  CHAR(32),
     FRESPACE   POSITION(xx)  INTEGER EXTERNAL(7),
     DELSPACE   POSITION(xx)  INTEGER EXTERNAL(1),
     DELCOUNT   POSITION(xx)  INTEGER EXTERNAL(1),
     FRAGIDX    POSITION(xx)  INTEGER EXTERNAL(1),
     MEDIATYP   POSITION(xx)  CHAR(2),
     CREDATE    POSITION(xx)  DATE EXTERNAL(10),
     ERRSTAT    POSITION(xx)  INTEGER EXTERNAL(1),
     VOLEMPTY   POSITION(xx)  CHAR(1),
     RECOUNT    POSITION(xx)  INTEGER EXTERNAL(1),
     CAPACITY   POSITION(xx)  INTEGER EXTERNAL(7),
     PLIBRARY   POSITION(xx)  CHAR(8),
     MEMBER     POSITION(xx)  CHAR(16))
 /*
//*********************************************************************
//CBRSMER5 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.deltable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.fordelo,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
```

Figure 94. CBRSMERG SAMPLIB Member (Part 6 of 7)

```
 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE DELOBJT
    (COLNAME    POSITION(xx)  CHAR(44),
     OBJNAME    POSITION(xx)  CHAR(44),
     VOLSER     POSITION(xx)  CHAR(6),
     VTOCTOKN   POSITION(xx)  INTEGER EXTERNAL(1),
     OBJSIZE    POSITION(xx)  INTEGER EXTERNAL(6))
 /*
//*********************************************************************
//CBRSMER6 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//          REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.tvoltble,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.fortvol,DISP=(MOD,KEEP),UNIT=3390,
//          VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.fortvol,DISP=(MOD,KEEP),UNIT=3390,
//          VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.fortvol,DISP=(MOD,KEEP),UNIT=3390,
//          VOL=SER=DBPACK
//SYSMAP   DD DSN=smerge.map,DISP=(MOD,KEEP),UNIT=3390,
//          VOL=SER=DBPACK
//SYSERR   DD DSN=smerge.err,DISP=(MOD,KEEP),UNIT=3390,
//          VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      RESUME YES
      INTO TABLE TAPEVOL
    (VOLSER     POSITION(xx)     CHAR(6),
     UNITNAME   POSITION(xx)     CHAR(8),
     MEDIATYP   POSITION(xx)     CHAR(2),
     STORGRP    POSITION(xx)     CHAR(8),
     TYPE       POSITION(xx)     CHAR(1),
     CREDATE    POSITION(xx)     DATE EXTERNAL(10),
     MNTDATE    POSITION(xx)     DATE EXTERNAL(10),
     WRTDATE    POSITION(xx)     DATE EXTERNAL(10),
     EXPDATE    POSITION(xx)     DATE EXTERNAL(10),
     CAPACITY   POSITION(xx)     INTEGER EXTERNAL(8)
     FRESPACE   POSITION(xx)     INTEGER EXTERNAL(8)
     LSTBLKID   POSITION(xx)     INTEGER EXTERNAL(8)
     PFULL      POSITION(xx)     INTEGER EXTERNAL(2),
     NUMLBLKS   POSITION(xx)     INTEGER EXTERNAL(8)
     NUMLKBW    POSITION(xx)     INTEGER EXTERNAL(8)
     NUMPKBW    POSITION(xx)     INTEGER EXTERNAL(8)
     NUMLKBDE   POSITION(xx)     INTEGER EXTERNAL(8)
     FULL       POSITION(xx)     CHAR(1),
     READABLE   POSITION(xx)     CHAR(1),
     WRITABLE   POSITION(xx)     CHAR(1),
     INUSE      POSITION(xx)     CHAR(1),
     COPIED     POSITION(xx)     CHAR(1),
     AVOLSER    POSITION(xx)     CHAR(6),
     COMPACT    POSITION(xx)     CHAR(1),
     MEMBER     POSITION(xx)     CHAR(16));
 /*
```

Figure 94. CBRSMERG SAMPLIB Member (Part 7 of 7)

CBRSG100

SAMPLIB member CBRSG100, as shown in Figure 95 on page 423, assists you in merging OAM administration databases and catalog entries, and OAM object storage databases, for use with DB2 data sharing in an OAMplex environment. The prerequisites must be performed and the job must be modified and run successfully before OAM is used.

```
//CBRSG100 JOB MSGLEVEL=(1,1),MSGCLASS=A
//*******************************************************************
//*
//*  $SEG(CBRSG100) COMP(DBM) PROD(OAM):
//*
//*  OAM Catalog Merge Job (combining collection name catalogs
//*  from multiple systems)
//*  OAM DB2 Admin Database Merge Job (combining OAM administration
//*  databases from multiple systems)
//*  OAM DB2 Object Database Merge Job (combining object directories
//*  and object storage databases from multiple systems)
//*
//*  ---------------------------------------------------------------
//*  ---------------------------------------------------------------
//*
//*  CBRSG100
//*
//*  This job is intended to assist in the task of merging OAM
//*  administration databases and catalog entries, and OAM object
//*  storage databases, for use with DB2 datasharing in an OAMplex
//*  environment.
//*  DFSMS 1.5.0 provides OAM support in a parallel sysplex
//*  environment using XCF communications and DB2 data sharing.
//*
//*  This job will:
//*   1. Load data from other system (DB2 data from other
//*      system's DB2 tables) into this system's OAM's tables.
//*
//******************************************************************
//*
//*  Before running this job, you MUST verify the following:
//*
//*    1. There are no two storage groups across any systems
//*       which are being combined, which have the same collection
//*       name associated with them. A collection CANNOT span object
//*       storage groups, therefore may belong to one and only one
//*       object storage group.
//*       If this condition exists:
//*          - the collection name on one of the systems being
//*            combined must change, and the ACS routines updated
//*            accordingly.
//*            OR
//*          - the two storage groups must be combined into a
//*            single storage group.
//*
//*    2. There are no two collection names across any systems
//*       that are being combined, which have the same collection
//*       ID.  Objects are associated with a collection by its ID,
//*       and the collection ID is unique in the collection table.
//*       Any collections across systems being combined, which had
//*       the same collection ID, must have been changed, and the
//*       object directory entries using these collections IDs,
//*       must have been udpated.
//*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 1 of 10)

```
//*    3. There are no two management classes across any systems
//*       that are being combined, which have the same management
//*       class ID.  Objects are associated with a management class
//*       in the object directory by its ID, and the management class
//*       ID is unique in the management class table.
//*       Any management classes across systems being combined which
//*       had the same ID, must have been changed, and the object
//*       directory entries using the modified management class' ID
//*       must have been updated.
//*
//*    4. There are no two storage classes across any systems
//*       that are being combined, which have the same storage
//*       class ID.  Objects are associated with a storage class
//*       in the object directory by its ID, and the storage class
//*       ID is unique in the storage class table.
//*       Any storage classes across systems being combined which
//*       had the same ID, must have been changed, and the object
//*       directory entries using the modified storage class' ID
//*       must have been updated.
//*
//*
//*   NOTE1:  In order to 'correct' a duplicate collection ID
//*           situation you can do the following:
//*           - it is best to make changes to the data that is
//*             being moved rather than the data on the system
//*             where the data is being combined.
//*           1. On the system where the data is to be combined,
//*              determine what the next available collection ID
//*              is that can be used.
//*           2. On the system where the data is coming from, with
//*              the duplicate collection ID, change the collection
//*              ID associated with the collection name in the
//*              collection table to the ID determined in step 1.
//*           3. Change all of the object directory entries in the
//*              object directory for the storage group to which the
//*              collection belongs, where the collection ID is the
//*              ID previously associated with the collection to
//*              the new collection ID used from step 2.
//*           4. Use IDCAMS to catalog the new collection in the
//*              catalog on the target system.
//*           example:
//*           Object directories on system 1 and system 2 are to be
//*           combined.  COLL.SYS1.DATA1997 on system 1 has the
//*           same collection ID as COLL.SYS2.DATA1997 on system 2.
//*
//*           BEFORE:
//*           -------
//*           System 1                       System 2
//*           Coll-name          Coll-ID     Coll-name          Coll-ID
//*
//*           OBJCOLL.SYS1.TEST   001        OBJCOLL.SYS2.TEST   001
//*           COLL.SYS1.DATA1997  002        COLL.SYS2.DATA1997  002
//*           COLL.SYS1.DATA1998  004        COLL.SYS2.DATA1998  003
//*           COLL.SYS1.REPORTS   005
//*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 2 of 10)

```
//*          System 1 will be the target system where data is combined
//*          OBJCOLL.SYS2.TEST will not be moved, as this is test data
//*          that is not needed on the combined system
//*
//*          Step 1:  The next available collection ID on
//*                   system 1 is 006
//*
//*          Step 2:  On System 2 before attempting the combine:
//*                   SQL UDPATE OAMADMIN.CBR_COLLECTION_TBL
//*                       SET ODCLID = 6 WHERE
//*                       ODCLNAME = 'COLL.SYS2.DATA1997';
//*                   COMMIT;
//*
//*          Step 3:  On System 2 before attempting the combine:
//*                   SQL UDPATE hlq.OSM_OBJ_DIR
//*                       SET ODCLID = 6 WHERE
//*                           ODCLID = 2;
//*                   COMMIT;
//*
//*          Step 4:  On System 1 before attempting the combine:
//*                   IDCAMS DEFINE NONVSAM RECATALOG -
//*                       COLLECTION NAME(COLL.SYS2.DATA1997)
//*
//*          AFTER:
//*          ------
//*          System 1                      System 2
//*          Coll-name          Coll-ID    Coll-name          Coll-ID
//*
//*          OBJCOLL.SYS1.TEST   001       OBJCOLL.SYS2.TEST   001
//*          COLL.SYS1.DATA1997  002       COLL.SYS2.DATA1997  006
//*          COLL.SYS1.DATA1998  004       COLL.SYS2.DATA1998  003
//*          COLL.SYS1.REPORTS   005
//*
//*
//* WARNING:  In order to 'correct' a duplicate collection name
//*          situation, you will have to investigate your SMS
//*          CDS more closely and see where the storage group is
//*          assigned based on the collection name and see if the
//*          data can be combined under one storage group, or if
//*          objects in one collection need to be changed to another
//*          collection altogether.  Then the ACS routines would have
//*          to also be updated to handle the new collection.
//*          This needs to be done with the assistance of your
//*          application interface and your systems programmer to
//*          determine the best plan for your installation.
//*
//*   NOTE2:  In order to 'correct' a duplicate management class
//*          number situation you can do similar steps as the ones
//*          to correct duplicate collection IDs, with the exception
//*          that the catalog step is not needed.
//*          In brief the steps would be:
//*          1. determine the next available management class number
//*             'x' below is the new managment class number
//*             'y' below is the old managment class number
//*          2. Change the management class number on the
//*             'from' system
//*             SQL UDPATE OAMADMIN.CBR_MGT_CLASS_TBL
//*                 SET MCNUM = x WHERE
//*                 ODMCNAME = 'duplicate.mc.name';
//*             COMMIT;
```

Figure 95. CBRSG100 SAMPLIB Member (Part 3 of 10)

```
//*          3. Change all of the object directory entries with the
//*             old management class number to the new management
//*             class number on the 'from' system
//*                   SQL UDPATE hlq.OSM_OBJ_DIR
//*                       SET ODMCNUM = x WHERE
//*                           ODMCNUM = y;
//*                   COMMIT;
//*
//*   NOTE3:  In order to 'correct' a duplicate storage class number
//*          situation you can do similar steps as the ones to
//*          correct duplicate collection IDs, with the exception
//*          that the catalog step is not needed.
//*          In brief the steps would be:
//*          1. determine the next available storage class number
//*             'x' below is the new storage class number
//*             'y' below is the old storage class number
//*          2. Change the storage class ID on the 'from' system
//*             SQL UDPATE OAMADMIN.CBR_STO_CLASS_TBL
//*                 SET SCNUM = x WHERE
//*                 ODMCNAME = 'duplicate.sc.name';
//*             COMMIT;
//*          3. Change all of the object directory entries with the
//*             old storage class number to the new storage class
//*             number on the 'from' system
//*                   SQL UDPATE hlq.OSM_OBJ_DIR
//*                       SET ODSCNUM = x WHERE
//*                           ODSCNUM = y;
//*                   COMMIT;
//*
//*******************************************************************
//*
//*  It is recommended that you create a DB2 image copy of the
//*  existing tables:
//*    OAM administration database tables:
//*       OAMADMIN.CBR_MGT_CLASS_TBL
//*       OAMADMIN.CBR_STO_CLASS_TBL
//*       OAMADMIN.CBR_COLLECTION_TBL
//*    OAM object directory and object storage database tables:
//*       all hlq.OSM_OBJ_DIR tables
//*       all hlq.OSM_04K_OBJ_TBL tables
//*       all hlq.OSM_32K_OBJ_TBL tables
//*
//*******************************************************************
//*
//*  Before running this job, you must change the following:
//*
//*    1.Change the PARM='DB2' in the JOB STEP statements to
//*      the name of the DB2 Subsystem in your installation.
//*
//*    2.Change the data set name SYS1.DB2.V4R1M0.SDSNLOAD
//*      in the STEPLIB statements to the data set name used
//*      for the DB2 SDNSLOAD dataset if necessary.
//*
//*    3.Change RESUME YES to RESUME NO if you are loading
//*      into empty tables.
//*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 4 of 10)

```
//*    4. Change the data set names in the job steps to the
//*       appropriate data set names for loading the DB2
//*       tables from other systems (this sample job is set up
//*       as though datasets are pre-allocated):
//*
//*       sg100.map       = map dataset for DB2 in the job (reused
//*                         in this job, or can use separate data
//*                         sets for each job step if desired
//*       sg100.err       = error dataset for DB2 in the job (reused
//*                         in this job, or can use separate data
//*                         sets for each job step if desired
//*
//*       input.clntable  = DSN with the collection table row values
//*                         from the system to be merged
//*       workdsn.forcln  = work dataset for DB2 in job step
//*       sortdsn.forcln  = sort dataset for DB2 in job step
//*       discard.forcln  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.mctable   = DSN with the management class table row
//*                         values from the system to be merged
//*       workdsn.formc   = work dataset for DB2 in job step
//*       sortdsn.formc   = sort dataset for DB2 in job step
//*       discard.formc   = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.sctable   = DSN with the storage class table row
//*                         values from the system to be merged
//*       workdsn.forsc   = work dataset for DB2 in job step
//*       sortdsn.forsc   = sort dataset for DB2 in job step
//*       discard.forsc   = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.objdir    = DSN with the object directory table row
//*                         values from the system to be merged
//*       workdsn.forodir = work dataset for DB2 in job step
//*       sortdsn.forodir = sort dataset for DB2 in job step
//*       discard.forodir = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.obj04k    = DSN with the object 4K table row
//*                         values from the system to be merged
//*       workdsn.for04k  = work dataset for DB2 in job step
//*       sortdsn.for04k  = sort dataset for DB2 in job step
//*       discard.for04k  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       input.obj32k    = DSN with the object 32K table row
//*                         values from the system to be merged
//*       workdsn.for32k  = work dataset for DB2 in job step
//*       sortdsn.for32k  = sort dataset for DB2 in job step
//*       discard.for32k  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*      **NOTE: For these datasets, use size calculations that
//*              would be needed for your installation, using the
//*              DB2 guidelines in the DB2 Command and Utility
//*              Reference for the LOAD utility.
//*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 5 of 10)

```
//*    5. Change storage_group_hlq in the job steps to the
//*       high level qualifier for the object storage group
//*       tables that are being merged.
//*
//*    6. Load the information from the OAM administration database
//*       tables and the object storage group databases tables from
//*       the other OAM/system where data is to be copied from into
//*       datasets on this system
//*       The simplest way to do this is to do an SQL SELECT * from
//*       the OAM tables to be merged on the other system and editing
//*       the column headers out of the resulting output so that just
//*       the data from the rows remains.
//*
//*    7. Note the beginning, (and ending if needed), columns
//*       where the data for each column resides in the table rows.
//*
//*    8. If necessary, change the POSITION(xx:yy) in the job
//*       steps to correlate to the actual beginning (and ending)
//*       columns where the data for each column resides in the
//*       input datasets (the SYSREC DD statmement dataset).
//*
//*       - For columns which are defined as specific length,
//*         like CHAR(2), only the start position is needed and
//*         the end will be determined by the length of the column
//*       - For columns which are defined as VARCHAR, only a
//*         start position is needed if the first 2 bytes of
//*         the data is the length of the following data.
//*         Otherwise, a beginning and ending designation are
//*         needed.
//*
//*    9. The integer fields are set up as EXTERNAL(zz) in the job
//*       steps, be sure that any integer values in the columns that
//*       are not the full length are padded with preceding zeroes
//*       in the input dataset (the SYSREC DD statement dataset).
//*
//*   10. If your input dataset (DD SYSREC) does not have the data
//*       always in specific columns because of the varying length
//*       fields, the easiest way to change the POSITION(xx:yy)
//*       statements would be to make the POSITION start column
//*       an offest from the end of the previous field.  For
//*       example:
//*
//*       LOAD DATA INDDN(SYSREC)
//*            REPLACE
//*            INTO TABLE OAMADMIN.CBR_COLLECTION_TBL
//*          (ODCLSCNM   POSITION(1 )  VARCHAR(30),
//*           ODCLMCNM   POSITION(+2)  VARCHAR(30),
//*           ODCLSGNM   POSITION(+2)  VARCHAR(30),
//*           ODCLID     POSITION(+2)     INTEGER,
//*           ODCLNAME   POSITION(+2)  VARCHAR(44));
//*
//*       In this example, the data for each subsequent column is
//*       expected to be 2 positions from the end of the preceding
//*       data, with the first 2 bytes of the VARCHAR fields being
//*       the actual length of the field's data.
//*
//*   11. Repeat steps CBRSG104-CBRSG106 for each set of object
//*       storage group tables that are being merged.
//*
//*******************************************************************
//*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 6 of 10)

```
//*  After running this job, do the following:
//*
//*    1. Check the return codes from the job to verify success
//*       or failure of the data merge.
//*
//*    2. Check the data sets below for any rows that could not
//*       be merged into the configuration database.  The most
//*       likely cause of failure would be duplicate rows.
//*
//*       discard.forcln  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       discard.formc   = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       discard.forsc   = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       discard.forodir = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       discard.for04k  = DSN for the output of rows that could
//*                         not be merged from the other system
//*
//*       discard.for32k  = DSN for the output of rows that could
//*                         not be merged from the other system
//*********************************************************************
//*  Load configuration tables from different DB2 database
//*********************************************************************
//CBRSG101 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.clntable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forcln,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forcln,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forcln,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE OAMADMIN.CBR_COLLECTION_TBL
    (ODCLSCNM   POSITION(xx)     VARCHAR,
     ODCLMCNM   POSITION(xx)     VARCHAR,
     ODCLSGNM   POSITION(xx)     VARCHAR,
     ODCLID     POSITION(xx)     INTEGER EXTERNAL(3),
     ODCLNAME   POSITION(xx)     VARCHAR)
/*
//*********************************************************************
```

Figure 95. CBRSG100 SAMPLIB Member (Part 7 of 10)

```
//CBRSG102 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.mctable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.formc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE OAMADMIN.CBR_MGT_CLASS_TBL
    (ODMCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODMCNAME   POSITION(xx)     VARCHAR)
 /*
//*******************************************************************
//CBRSG103 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.sctable,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forsc,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE OAMADMIN.CBR_STO_CLASS_TBL
    (ODSCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODSCNAME   POSITION(xx:yy)  VARCHAR)
 /*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 8 of 10)

```
//*********************************************************************
//*  Combine hlq.OSM_OBJ_DIR tables from different
//*  DB2 systems for object storage group.
//*********************************************************************
//CBRSG104 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.objdir,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.forodir,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE storage_group_hlq.OSM_OBJ_DIR
    (ODVER      POSITION(xx)     CHAR(1),
     ODSIZE     POSITION(xx)     INTEGER EXTERNAL(3),
     ODCREATS   POSITION(xx)     TIMESTAMP EXTERNAL(26),
     ODEXPDT    POSITION(xx)     DATE EXTERNAL(10),
     ODLREFDT   POSITION(xx)     DATE EXTERNAL(10),
     ODPENDDT   POSITION(xx)     DATE EXTERNAL(10),
     ODMCASDT   POSITION(xx)     DATE EXTERNAL(10),
     ODSCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODMCNUM    POSITION(xx)     INTEGER EXTERNAL(2),
     ODLOCFL    POSITION(xx)     CHAR(1),
     ODLSLOC    POSITION(xx)     CHAR(6),
     ODSECLOC   POSITION(xx)     INTEGER EXTERNAL(4),
     ODBKLOC    POSITION(xx)     CHAR(6),
     ODBKSEC    POSITION(xx)     INTEGER EXTERNAL(4),
     ODCLID     POSITION(xx)     INTEGER EXTERNAL(4),
     ODNAME     POSITION(xx:yy)  VARCHAR)
/*
//*********************************************************************
//*  Combine hlq.OSM_04K_OBJ_DIR tables from different
//*  DB2 systems for object storage group.
//*********************************************************************
//CBRSG105 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.obj04k,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.for04k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.for04k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.for04k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
```

Figure 95. CBRSG100 SAMPLIB Member (Part 9 of 10)

```
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE storage_group_hlq.OSM_04K_OBJ_TBL
    (OTVER      POSITION(xx)     CHAR(1),
     OTSEG      POSITION(xx)     INTEGER EXTERNAL(2),
     OTCLID     POSITION(xx)     INTEGER EXTERNAL(3),
     OTNAME     POSITION(xx:yy)  VARCHAR,
     OTOBJ      POSITION(xx:yy)  VARCHAR)
/*
//*******************************************************************
//*  Combine hlq.OSM_32K_OBJ_DIR tables from different
//*  DB2 systems for object storage group.
//*******************************************************************
//CBRSG106 EXEC PGM=DSNUTILB,DYNAMNBR=20,COND=(4,LT),PARM='DB2',
//         REGION=4096K
//STEPLIB  DD DSN=SYS1.DB2.V4R1M0.SDSNLOAD,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSREC   DD DSN=input.obj32k,DISP=(OLD,KEEP)
//SYSUT1   DD DSN=workdsn.for32k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SORTOUT  DD DSN=sortdsn.for32k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSDISC  DD DSN=discard.for32k,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSMAP   DD DSN=sg100.map,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSERR   DD DSN=sg100.err,DISP=(MOD,KEEP),UNIT=3390,
//         VOL=SER=DBPACK
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *

 LOAD DATA INDDN(SYSREC)
      REPLACE
      INTO TABLE storage_group_hlq.OSM_32K_OBJ_TBL
    (OTVER      POSITION(xx)     CHAR(1),
     OTSEG      POSITION(xx)     INTEGER EXTERNAL(2),
     OTCLID     POSITION(xx)     INTEGER EXTERNAL(3),
     OTNAME     POSITION(xx:yy)  VARCHAR,
     OTOBJ      POSITION(xx:yy)  VARCHAR)
/*
```

Figure 95. CBRSG100 SAMPLIB Member (Part 10 of 10)

Application Plans

DB2 BIND (CBRxBIND) and GRANT (CBRxGRNT) jobs are provided to create and authorize the DB2 application plans necessary for OSR, OSMC, LCS, and ISMF.

CBRPBIND

CBRPBIND performs a DB2 BIND for the packages that are needed to access the OAM object storage group tables. The use of DB2 packages allows user defined qualifiers for the object storage group table definitions. For release-to-release consistency, the job provides binds packages for 100 object storage groups (GROUP00–GROUP99). With the DFSMS 1.5.0 support, the 100 Object storage group restriction is no longer valid, so this job must be modified to match your installation requirements. You must run this job prior to the execution of CBRABIND, CBRHBIND, or CBRIBIND.

CBRIBIND and CBRIGRNT

The CBRIBIND and CBRIGRNT SAMPLIB jobs create the OSR application plan, bind it to DB2, and grant authority for the plan to be used. Run these SAMPLIB jobs if you do not plan to create the optical configuration database, but you do plan to store objects without starting the OAM address space. If you plan on creating the optical configuration database and start the OAM address space, use CBRABIND and CBRAGRNT in place of these SAMPLIB members.

CBRHBIND and CBRHGRNT

The CBRHBIND and CBRHGRNT SAMPLIB jobs create the OSMC application plans, bind them to DB2, and grant authority for the plans to be used. Run these SAMPLIB jobs if you plan to start the OAM address space with OSMC.

CBRABIND and CBRAGRNT

The CBRABIND and CBRAGRNT SAMPLIB jobs create the LCS, OSR, and ISMF application plans, bind them to DB2, and grant authority for the plans to be used. Run these SAMPLIB jobs if you plan to create the optical configuration database and start the OAM address space for object support.

OAM Installation Verification Program and OAMUTIL

Sample jobs are provided for the invocation of the OAM installation verification program using TSO and the OAM utility, OAMUTIL, to reformat 3995 optical cartridges.

OAMIVP

The OSREQ macro is a TSO command processor. It can be invoked for any of the following situations:

- At the TSO READY prompt when logged on to TSO
- Under option 6 “Command” under ISPF/PDF
- On the ISPF command line by prefacing the OSREQ command with the characters “TSO”
- From within a TSO CLIST
- From a batch job, by invoking the TSO terminal monitor program (TMP) in the batch job

SAMPLIB member OAMIVP invokes the OSREQ macro from an MVS batch job, which invokes the TSO terminal monitor program (IKKJEFT01) that runs in the background. This job allows you to verify the installation of OAM to ensure object storage success.

```
//OAMIVP   JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//*********************************************************************
//*                                                                   *
//*  $SEG(CBRSAMIV) COMP(OSR) PROD(OAM):                              *
//*                                                                   *
//* THIS SAMPLE JOB INVOKES THE OSREQ COMMAND PROCESSOR TO PERFORM    *
//* THE FOLLOWING ACTIONS:                                            *
//*                                                                   *
//* 1. STORE A 1 MB (1048576) OBJECT.                                 *
//*                                                                   *
//* 2. ISSUE AN OSREQ QUERY TO LIST THE OAM DIRECTORY INFORMATION.    *
//*                                                                   *
//* 3. ISSUE A LISTCAT COMMAND TO LIST THE COLLECTION NAME ENTRY      *
//*    FROM THE ICF CATALOG.                                          *
//*                                                                   *
//* 4. ISSUE AN OSREQ CHANGE REQUEST TO CHANGE THE RETENTION PERIOD   *
//*    ASSOCIATED WITH THE OBJECT TO 365 DAYS.                        *
//*                                                                   *
//* 5. ISSUE AN OSREQ QUERY TO LIST THE OAM DIRECTORY INFORMATION.    *
//*                                                                   *
//* 6. ISSUE AN OSREQ RETRIEVE TO RETRIEVE THE PRIMARY COPY OF THE    *
//*    AND CHECK TO SEE IF IT CONTAINS THE PRE-DEFINED PATTERN DATA.  *
//*                                                                   *
//* 7. ISSUE AN OSREQ DELETE REQUEST TO DELETE THE OBJECT.            *
//*                                                                   *
//*                                                                   *
//* PRIOR TO EXECUTING THIS JOB YOU MAY NEED TO MAKE THE FOLLOWING    *
//* MODIFICATIONS:                                                    *
//*                                                                   *
//*  1. CHANGE THE JOB CARD PER YOUR INSTALLATION REQUIREMENTS.       *
//*                                                                   *
//*  2. CHANGE THE COLLECTION NAME, OBJECT NAME AND LENGTH OF         *
//*     OBJECT TO CONFORM TO YOUR INSTALLATION REQUIREMENTS.          *
//*                                                                   *
//*********************************************************************
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OSREQ STORE    COLLECT.NAME OBJECT.NAME LENGTH(1048576)
 OSREQ QUERY    COLLECT.NAME OBJECT.NAME
 LISTCAT ENTRIES('COLLECT.NAME') ALL
 OSREQ CHANGE   COLLECT.NAME OBJECT.NAME RP(365)
 OSREQ QUERY    COLLECT.NAME OBJECT.NAME
 OSREQ RETRIEVE COLLECT.NAME OBJECT.NAME COMPARE VIEW(PRIMARY)
 OSREQ DELETE   COLLECT.NAME OBJECT.NAME
/*
/*
```

Figure 96. OAMIVP SAMPLIB Member

CBRSAMUT

SAMPLIB member CBRSAMUT, as shown in Figure 97 on page 435, uses the OAMUTIL utility to reformat a 3995 optical disk volume or both sides of a 3995 optical cartridge. For more information on this utility command, see “Reformatting a 3995 Optical Disk” on page 266.

```
//OAMUTIL  JOB CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//*******************************************************************
//*                                                                 *
//*  $SEG(CBRSAMUT) COMP(LCS) PROD(OAM):                            *
//*                                                                 *
//*                                                                 *
//* This sample job provides examples of invoking OAMUTIL command   *
//*      processor to reformat the 3995 optical disk cartridge.     *
//*                                                                 *
//* Following are some examples:                                    *
//*                                                                 *
//* example #01- conditionally reformat one side:                   *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1                             *
//*                                                                 *
//* example #02- conditionally reformat one side on a specific      *
//*              operator accessible drive:                         *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 DRIVENAME(P21D5)            *
//*                                                                 *
//* example #03- conditionally reformat one side and rename it      *
//*              to a new volume serial number:                     *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 NEWVOL1(newvs1)             *
//*                                                                 *
//* example #04- Conditionally reformat both sides:                 *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH                        *
//*                                                                 *
//* example #05- Conditionally reformat both sides and              *
//*              return them to SCRATCH storage group:              *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH SCRATCH                *
//*                                                                 *
//* example #06- Conditionally reformat both sides on               *
//*              a specific operator accessible drive:              *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH DRIVENAME(P52D1)       *
//*                                                                 *
//* example #07- Conditionally reformat both sides and              *
//*              rename them to new volume serial numbers:          *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2)  *
//*                                                                 *
//* example #08- Conditionally reformat both sides,                 *
//*              return them to SCRATCH storage group, and          *
//*              rename them to new volume serial numbers:          *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)          + *
//*                                  NV2(newvs2) SCRATCH            *
//*                                                                 *
//* example #09- Conditionally reformat both sides on               *
//*              a specific operator accessible drive,              *
//*              return them to SCRATCH storage group, and          *
//*              rename them to new volume serial numbers:          *
//*                                                                 *
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)          + *
//*                             NV2(newvs2) D(P52D1) SCRATCH  *
//*                                                                 *
```

Figure 97. CBRSAMUT SAMPLIB Member (Part 1 of 2)

```
//* example #10- Unconditionally reformat one side:                *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 FORCE                      *
//*                                                                *
//* example #11- Unconditionally reformat one side on a specific   *
//*              operator accessible drive:                        *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 DRIVENAME(P21D5) FORCE     *
//*                                                                *
//* example #12- Unconditionally reformat one side and rename it   *
//*              to a new volume serial number:                    *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 NEWVOL1(newvs1) FORCE      *
//*                                                                *
//* example #13- Unconditionally reformat both sides:              *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 BOTH FORCE                 *
//*                                                                *
//* example #14- Unconditionally reformat both sides and           *
//*              return them to SCRATCH storage group:             *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 BOTH SCRATCH FORCE         *
//*                                                                *
//* example #15- Unconditionally reformat both sides on            *
//*              a specific operator accessible drive:             *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 BOTH DRIVENAME(P52D1) FORCE   *
//*                                                                *
//* example #16- Unconditionally reformat both sides and           *
//*              rename them to new volume serial numbers:         *
//*                                                                *
//*       OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2) FORCE  *
//*                                                                *
//* example #17- Unconditionally reformat both sides,              *
//*              return them to SCRATCH storage group, and         *
//*              rename them to new volume serial numbers:         *
//*                                                                *
//*             OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1)         + *
//*                                         NV2(newvs2) SCRATCH FORCE *
//*                                                                *
//*                                                                *
//* PRIOR TO EXECUTING THIS JOB YOU MAY NEED TO MAKE THE FOLLOWING    *
//* MODIFICATIONS:                                                 *
//*                                                                *
//*  1. CHANGE THE JOB CARD PER YOUR INSTALLATION REQUIREMENTS.     *
//*                                                                *
//*  2. CHANGE THE VOLUME SERAL NUMBER(S), OPERATOR ACCESSIBLE      *
//*     DRIVE  TO CONFORM TO YOUR INSTALLATION REQUIREMENTS.        *
//*                                                                *
//*******************************************************************
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OAMUTIL REFORMAT oldvs1 NV1(newvs1)
 OAMUTIL REFORMAT oldvs1 BOTH NV1(newvs1) NV2(newvs2) D(P52D1)
 OAMUTIL REFORMAT oldvs1                                            +
                  NV1(newvs1) NV2(newvs2) D(P21D5) SCRATCH FORCE
/*
```

Figure 97. CBRSAMUT SAMPLIB Member (Part 2 of 2)

Automatic Class Selection

SAMPLIB members CBRHSC, CBRHMC, and CBRHSG are sample automatic class selection routines for the OAM environments.

CBRHSC

SAMPLIB member CBRHSC, as shown in Figure 98, provides a storage class ACS routine for STORE, CHANGE, and CTRANS.

```
/*********************************************************************/
/*                                                                   */
/* $SEG(CBRHSC) COMP(OSMC) PROD(OAM):                                */
/*                                                                   */
/* OAM Sample Storage Class ACS Routine                              */
/*                                                                   */
/* FUNCTION:SUPPLY A STORAGE CLASS FOR OAM OBJECTS                   */
/*                                                                   */
/* OPERATION:Supply a storage class for the following environments:  */
/*                                                                   */
/*             STORE   - Assign initial storage class of DASD or     */
/*                       optical library based on collection name.   */
/*             CHANGE  - The storage class of an object has been     */
/*                       requested to change.                        */
/*             CTRANS  - The object is moved in the hierarchy        */
/*                       according to management class.              */
/*                                                                   */
/* NOTES:    In this implementation, the collection name is used as  */
/*            the basis for determining whether explicit values will */
/*            be considered. This approach and all of the processing */
/*            indicated below is one of many differing possibilities */
/*            and is only for purposes of illustration to demonstrate*/
/*            the types of processing that can be accomplished in an */
/*            ACS routine. Actual implementations will vary.         */
/*                                                                   */
/* ASSUMPTIONS:                                                      */
/*   Collection name format                                          */
/*     CLLCT0mn                                                      */
/*       where m = 0 or 1 (0 indicates that explicit values will be  */
/*                         ignored, 1 indicates that explicit values */
/*                         are considered in some cases and may      */
/*                         result in an override)                    */
/*       where n = 0 - 9                                             */
/*   Object name format - has 5 levels as follows:                   */
/*     xxxx.xxx.xxxxxxxx.xxxxxxx.xxx                                 */
/*   Valid storage classes:                                          */
/*     DB2DASD  - DASD                                               */
/*     OLIBRARY - optical                                            */
/*     TAPESC   - tape                                               */
/*                                                                   */
```

Figure 98. CBRHSC SAMPLIB Member (Part 1 of 5)

```
/********************************************************************/
/*            L O G I C       O V E R V I E W                      */
/*                                                                  */
/* If STORE environment                                             */
/* | If object name not specified (i.e. this is an invocation for   */
/* | the entire collection)                                         */
/* | | Select                                                       */
/* | | When the collection is in the set that we are defining to    */
/* | | allow overrides                                              */
/* | | | If storage class specified is not 'OLIBRARY'               */
/* | | | | Set the storage class to 'DB2DASD'                       */
/* | | | Endif                                                      */
/* | | When the collection is in the set that we are defining to    */
/* | | not allow overrides                                          */
/* | | | Set the storage class to 'OLIBRARY'                        */
/* | | Otherwise                                                    */
/* | | | Set error code                                             */
/* | | End                                                          */
/* | Else (an object name was specified)                            */
/* | | Select                                                       */
/* | | When the collection is in the set that we are defining to    */
/* | | allow overrides                                              */
/* | | | If storage class specified is not 'OLIBRARY'               */
/* | | | | Set the storage class to 'DB2DASD'                       */
/* | | | Endif                                                      */
/* | | When the collection is in the set that we are defining to    */
/* | | not allow overrides                                          */
/* | | | If the object name has exactly 5 levels and the 5th        */
/* | | | level indicates that the object should have a particular   */
/* | | | storage class                                              */
/* | | | | Set the storage class to 'DB2DASD'                       */
/* | | | Else                                                       */
/* | | | | Set the storage class to 'OLIBRARY'                      */
/* | | | Endif                                                      */
/* | | Otherwise                                                    */
/* | | | Set error code                                             */
/* | | End                                                          */
/* | Endif (object name specified)                                  */
/* Endif (STORE environment)                                        */
/*                                                                  */
/*                                                                  */
/* If CHANGE environment                                            */
/* | If the storage class specified is not a valid storage class    */
/* | | Set error code                                               */
/* | Endif                                                          */
/* Endif (CHANGE environment)                                       */
/*                                                                  */
/*                                                                  */
/* If CLASS_TRANSITION environment                                  */
/* | Select                                                         */
/* | When storage class is 'DB2DASD'                                */
/* | | Set storage class to 'OLIBRARY'                              */
/* | When storage class is 'OLIBRARY'                               */
/* | | Set storage class to 'TAPESC'                                */
/* | Otherwise                                                      */
/* | | Set error code                                               */
/* | End                                                            */
/* Endif                                                            */
```

Figure 98. CBRHSC SAMPLIB Member (Part 2 of 5)

```
/*******************************************************************/
/*     S T O R A G E   C L A S S   D E F I N I T I O N S         */
/*                                                                 */
/*Relevant                                                         */
/*Fields    DB2DASD OLIBRARY TAPESC                                */
/*--------- ------- -------- -------                               */
/*INITIAL                                                          */
/*ACCESS                                                           */
/*RESPONSE                                                         */
/*SECONDS       0      20      900                                 */
/*                                                                 */
/*SUSTAINED                                                        */
/*DATA                                                             */
/*RATE         n/a      1        3                                 */
/*                                                                 */
/*******************************************************************/


PROC STORCLAS                       /* Select a storage class     */

  FILTLIST BLANK    INCLUDE ('        ','')

/*******************************************************************/
/* STORE                                                           */
/*******************************************************************/
 IF &ACSENVIR = 'STORE' THEN          /* Object is being stored     */
   IF &MEMN = &BLANK THEN             /* If the object name is not
                                         specified (i.e. indicating
                                         an invocation for the entire
                                         collection)                */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
         | &DSN = 'CLLCT019')         /* For the collections which
                                         allow overrides            */
       IF &STORCLAS ¬= 'OLIBRARY' THEN /* If the storage class
                                         specified is not the one
                                         explicit storage class value
                                         that is considered valid
                                         for these collections      */
         SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                         desired value for these
                                         collections                */
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
         | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
         | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
         | &DSN = 'CLLCT009')         /* For the collections which
                                         do not allow overrides     */
       SET &STORCLAS = 'OLIBRARY'      /* Set the storage class to the
                                         desired value for these
                                         collections                */
     OTHERWISE                        /* Otherwise the collection name
                                         is invalid                 */
       EXIT CODE(10)                  /* Indicate that an error
                                         occurred                   */
     END                              /* Select                     */
```

Figure 98. CBRHSC SAMPLIB Member (Part 3 of 5)

```
  ELSE                                /* If the object name is
                                         specified (i.e. indicating
                                         an invocation for the
                                         specific collection)      */
    SELECT
    WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
        | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
        | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
        | &DSN = 'CLLCT019')          /* For the collections which
                                         allow overrides           */
      IF &STORCLAS ¬= 'OLIBRARY' THEN /* If the storage class
                                         specified is not the one
                                         explicit storage class value
                                         that is considered valid
                                         for these collections     */
        SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                         desired value for these
                                         collections               */
    WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
        | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
        | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
        | &DSN = 'CLLCT009')          /* For the collections which
                                         do not allow overrides    */
      IF &MEMN = *.*.*.*.DZX THEN     /* If the object name contains
                                         a value indicating that this
                                         object should be treated
                                         differently than the
                                         other objects in these
                                         collections               */
        SET &STORCLAS = 'DB2DASD'     /* Set the storage class to the
                                         desired value for these
                                         objects that are treated
                                         differently               */
      ELSE                            /* Otherwise there is nothing
                                         special about this object
                                         name                      */
        SET &STORCLAS = 'OLIBRARY'    /* Set the storage class to the
                                         desired value for the objects
                                         in these collections      */
    OTHERWISE                         /* Otherwise the collection name
                                         is invalid                */
      EXIT CODE(11)                   /* Indicate that an error
                                         occurred                  */
    END                               /* Select                    */

/*********************************************************************/
/* CHANGE                                                            */
/*********************************************************************/
 IF &ACSENVIR = 'CHANGE' THEN         /* Object is being changed    */
   IF &STORCLAS ¬= 'DB2DASD'  AND
      &STORCLAS ¬= 'OLIBRARY' AND
      &STORCLAS ¬= 'TAPESC'  THEN     /* If the storage class specified
                                         is not a storage class that
                                         is considered valid       */
       EXIT CODE(12)                  /* Indicate that an error
                                         occurred                  */
```

Figure 98. CBRHSC SAMPLIB Member (Part 4 of 5)

```
/********************************************************************/
/* CLASS TRANSITION                                                 */
/********************************************************************/
 IF &ACSENVIR = 'CTRANS' THEN          /* Object is being processed
                                          as a result of a class
                                          transition                */
   SELECT (&STORCLAS)

   WHEN ('DB2DASD')                    /* If current storage class
                                          indicates that object is on
                                          DASD                      */
     SET &STORCLAS = 'OLIBRARY'        /* Set storage class to indicate
                                          that the object should reside
                                          on optical                */
   WHEN ('OLIBRARY')                   /* If current storage class
                                          indicates that object is on
                                          optical                   */
     SET &STORCLAS = 'TAPESC'          /* Set storage class to indicate
                                          that the object should reside
                                          on tape                   */

   OTHERWISE                           /* Otherwise the storage class
                                          is invalid                */
     EXIT CODE(13)                     /* Indicate that an error
                                          occurred                  */
   END
END
```

Figure 98. CBRHSC SAMPLIB Member (Part 5 of 5)

CBRHMC

SAMPLIB member CBRHMC provides a management class ACS routine for OAM objects.

```
/*******************************************************************/
/*                                                                 */
/* $SEG(CBRHMC) COMP(OSMC) PROD(OAM):                              */
/*                                                                 */
/* OAM Sample Management Class ACS Routine                         */
/*                                                                 */
/* FUNCTION:SUPPLY A MANAGEMENT CLASS FOR OAM OBJECTS              */
/*                                                                 */
/*  OPERATION:Supply a management class for the following          */
/*             environments:                                       */
/*                                                                 */
/*               STORE   - Assign an initial management class based  */
/*                         on collection name and/or object name     */
/*                         and/or storage class.                   */
/*               CHANGE  - Validate a request to change the        */
/*                         management class for an object.         */
/*               CTRANS  - The object is moved in the hierarchy    */
/*                         according to its previous management    */
/*                         class.                                  */
/*                                                                 */
/* NOTES:    In this implementation, the collection name is used as  */
/*            the basis for determining whether explicit values will */
/*            be considered. This approach and all of the processing */
/*            indicated below is one of many differing possibilities */
/*            and is only for purposes of illustration to demonstrate*/
/*            the types of processing that can be accomplished in an */
/*            ACS routine. Actual implementations will vary.       */
/*                                                                 */
/* ASSUMPTIONS:                                                    */
/*   Collection name format                                        */
/*     CLLCT0mn                                                    */
/*       where m = 0 or 1 (0 indicates that explicit values will be  */
/*                         ignored, 1 indicates that explicit values */
/*                         are considered in some cases and may    */
/*                         result in an override)                  */
/*       where n = 0 - 9                                           */
/*   Object name format - has 5 levels as follows:                 */
/*     xxxx.xxx.xxxxxxxx.xxxxxxx.xxx                               */
/*   Valid storage classes:                                        */
/*     DB2DASD  - DASD                                             */
/*     OLIBRARY - optical                                          */
/*     TAPESC   - Tape                                             */
/*   Valid management classes:                                     */
/*     MAGONLY  - 30 days on DASD, then expire                     */
/*     MAG30D   - 30 days on DASD, then transition                 */
/*     MAG30LIB - 6 months on optical, then transition             */
/*     TAPSEVEN - 7 years on tape, then expire                     */
/*     OPT6D    - 0 days on DASD, then transition                  */
/*     OPT6LIB  - 6 months on optical, then transition             */
/*     OPTTAPE  - 7 years on tape, the expire                      */
/*                                                                 */
```

Figure 99. CBRHMC SAMPLIB Member (Part 1 of 8)

```
/*********************************************************************/
/*            L O G I C       O V E R V I E W                      */
/*                                                                   */
/* If STORE environment                                              */
/* | If object name not specified (i.e. this is an invocation for   */
/* | the entire collection)                                          */
/* | | Select                                                        */
/* | | When the collection is in the set that we are defining to     */
/* | | allow overrides                                               */
/* | | | Select                                                      */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)             */
/* | | | | If management class specified is not 'MAGONLY'            */
/* | | | | | Set the management class to 'MAG30D'                    */
/* | | | | Endif                                                     */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)         */
/* | | | | Set the management class to 'OPT6D'                       */
/* | | | Otherwise                                                   */
/* | | | | Set error code                                            */
/* | | | End                                                         */
/* | | When the collection is in the set that we are defining to     */
/* | | not allow overrides                                           */
/* | | | Select                                                      */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)             */
/* | | | | Set the management class to 'MAG30D'                      */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)         */
/* | | | | Set the management class to 'OPT6D'                       */
/* | | | Otherwise                                                   */
/* | | | | Set error code                                            */
/* | | | End                                                         */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Else (an object name was specified)                             */
/* | | Select                                                        */
/* | | When the collection is in the set that we are defining to     */
/* | | allow overrides                                               */
/* | | | If the object name has exactly 5 levels and the 5th         */
/* | | | level indicates that the object may have the management     */
/* | | | class overridden and the storage class is 'DB2DASD'         */
/* | | | | If management class specified is not 'MAGONLY'            */
/* | | | | | Set the management class to 'MAG30D'                    */
/* | | | | Endif                                                     */
/* | | | Else                                                        */
/* | | | | Select                                                    */
/* | | | | When the storage class is 'DB2DASD' (i.e. DASD)           */
/* | | | | | Set the management class to 'MAG30D'                    */
/* | | | | When the storage class is 'OLIBRARY' (i.e. Optical)       */
/* | | | | | Set the management class to 'OPT6D'                     */
/* | | | | Otherwise                                                 */
/* | | | | | Set error code                                          */
/* | | | | End                                                       */
/* | | | Endif                                                       */
/* | | When the collection is in the set that we are defining to     */
/* | | not allow overrides                                           */
/* | | | Select                                                      */
/* | | | When the storage class is 'DB2DASD' (i.e. DASD)             */
/* | | | | Set the management class to 'MAG30D'                      */
/* | | | When the storage class is 'OLIBRARY' (i.e. Optical)         */
/* | | | | Set the management class to 'OPT6D'                       */
/* | | | Otherwise                                                   */
/* | | | | Set error code                                            */
/* | | | End                                                         */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Endif (object name specified)                                   */
/* Endif (STORE environment)                                         */
/*                                                                   */
```

Figure 99. CBRHMC SAMPLIB Member (Part 2 of 8)

```
/*                                                                   */
/* If CHANGE environment                                             */
/* | Select                                                          */
/* | When storage class is 'DB2DASD'                                 */
/* | | If management class is not 'MAGONLY' or 'MAG30D'              */
/* | | | Set management class to 'MAG30D'                            */
/* | | Endif                                                         */
/* | When storage class is 'OLIBRARY'                                */
/* | | Set management class to 'OPT6LIB'                             */
/* | When storage class is 'TAPESC'                                  */
/* | | Set management class to 'OPTTAPE'                             */
/* | Otherwise                                                       */
/* | | Set error code                                                */
/* | End                                                             */
/* Endif (CHANGE environment)                                        */
/*                                                                   */
/*                                                                   */
/* If CLASS_TRANSITION environment                                   */
/* | Select                                                          */
/* | When storage class is 'OLIBRARY'                                */
/* | | Select                                                        */
/* | | When management class is 'OPT6D'                              */
/* | | | Set management class to 'OPT6LIB'                           */
/* | | When management class is 'MAG30D'                             */
/* | | | Set management class to 'MAG30LIB'                          */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | When storage class is 'TAPESC'                                  */
/* | | Select                                                        */
/* | | When management class is 'OPT6LIB'                            */
/* | | | Set management class to 'OPTTAPE'                           */
/* | | When management class is 'MAG30LIB'                           */
/* | | | Set management class to 'TAPSEVEN'                          */
/* | | Otherwise                                                     */
/* | | | Set error code                                              */
/* | | End                                                           */
/* | Otherwise                                                       */
/* | | Set error code                                                */
/* | End                                                             */
/* Endif                                                             */
/*********************************************************************/
/*  M A N A G E M E N T    C L A S S    D E F I N I T I O N S        */
/*                                                                   */
/*Relevant                                                           */
/*Fields    MAGONLY MAG30D  MAG30LIB TAPSEVEN OPT6D   OPT6LIB OPTTAPE*/
/*--------- ------- ------- -------- -------- ------- ------- -------*/
/*EXPIRE                                                             */
/*AFTER                                                              */
/*DAYS                                                               */
/*NON-USAGE NOLIMIT NOLIMIT NOLIMIT  NOLIMIT  NOLIMIT NOLIMIT NOLIMIT*/
/*                                                                   */
/*EXPIRE                                                             */
/*AFTER                                                              */
/*DATE/DAYS    30     2557    2557     2557     2557    2557    2557 */
/*                                                                   */
/*MAXIMUM                                                            */
/*RETENTION                                                          */
/*PERIOD       30   NOLIMIT NOLIMIT  NOLIMIT  NOLIMIT NOLIMIT NOLIMIT*/
/*                                                                   */
/*AUTO                                                               */
/*BACKUP      YES      NO      NO       NO       NO     YES      NO  */
/*                                                                   */
```

Figure 99. CBRHMC SAMPLIB Member (Part 3 of 8)

```
/*TIME                                                             */
/*SINCE                                                            */
/*CREATION                                                         */
/*YEARS       --      --      --       07      --      --      07  */
/*                                                                 */
/*TIME                                                             */
/*SINCE                                                            */
/*CREATION                                                         */
/*MONTHS      --      --      06       --      --      06      --  */
/*                                                                 */
/*TIME                                                             */
/*SINCE                                                            */
/*CREATION                                                         */
/*DAYS        --      30      --       --      00      --      --  */
/*                                                                 */
/*******************************************************************/

PROC MGMTCLAS                          /* Select an Management class */

   FILTLIST BLANK   INCLUDE ('        ','')

/*******************************************************************/
/* STORE                                                           */
/*******************************************************************/
 IF &ACSENVIR = 'STORE' THEN           /* Object is being stored     */
   IF &MEMN = &BLANK THEN              /* If the object name is not
                                          specified (i.e. indicating
                                          an invocation for the entire
                                          collection)                */
     SELECT
     WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
         | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
         | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
         | &DSN = 'CLLCT019')          /* For the collections which
                                          allow overrides            */
       SELECT (&STORCLAS)

       WHEN ('DB2DASD')                /* If current storage class
                                          indicates that object is on
                                          DASD                       */
         IF &MGMTCLAS ¬= 'MAGONLY' THEN /* If the specified management
                                          class value is not a valid
                                          override                   */
           SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                          indicate the DASD management
                                          specifications             */
       WHEN ('OLIBRARY')               /* If current storage class
                                          indicates that object is on
                                          optical                    */
         SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                          indicate the optical
                                          management specifications  */
       OTHERWISE                       /* Otherwise the storage class
                                          is invalid                 */
         EXIT CODE(20)                 /* Indicate that an error
                                          occurred                   */
       END
```

Figure 99. CBRHMC SAMPLIB Member (Part 4 of 8)

```
    WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
        | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
        | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
        | &DSN = 'CLLCT009')          /* For the collections which
                                         do not allow overrides     */
      SELECT (&STORCLAS)

      WHEN ('DB2DASD')                /* If current storage class
                                         indicates that object is on
                                         DASD                       */
        SET &MGMTCLAS = 'MAG30D'      /* Set management class to
                                         indicate the DASD management
                                         specifications             */
      WHEN ('OLIBRARY')               /* If current storage class
                                         indicates that object is on
                                         optical                    */
        SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                         indicate the optical
                                         management specifications  */
      OTHERWISE                       /* Otherwise the storage class
                                         is invalid                 */
        EXIT CODE(21)                 /* Indicate that an error
                                         occurred                   */
      END
    OTHERWISE                         /* Otherwise the collection name
                                         is invalid                 */
      EXIT CODE(22)                   /* Indicate that an error
                                         occurred                   */
    END                               /* Select                     */
  ELSE                                /* If the object name is
                                         specified (i.e. indicating
                                         an invocation for the
                                         specific collection)       */
    SELECT
    WHEN (&DSN = 'CLLCT010' | &DSN = 'CLLCT011' | &DSN = 'CLLCT012'
        | &DSN = 'CLLCT013' | &DSN = 'CLLCT014' | &DSN = 'CLLCT015'
        | &DSN = 'CLLCT016' | &DSN = 'CLLCT017' | &DSN = 'CLLCT018'
        | &DSN = 'CLLCT019')          /* For the collections which
                                         allow overrides            */
      IF &MEMN = *.*.*.*.IAX AND
         &STORCLAS = 'DB2DASD' THEN   /* If the object name contains
                                         a value indicating that this
                                         object should be treated
                                         differently than the
                                         other objects in these
                                         collections (i.e. only
                                         specific objects within
                                         these collections allow
                                         overrides) and the storage
                                         class indicates that the
                                         object is on DASD          */
        IF &MGMTCLAS ¬= 'MAGONLY' THEN /* If the specified management
                                         class value is not a valid
                                         override                   */
          SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                         indicate the DASD management
                                         specifications             */
        ELSE
          DO
          END
```

Figure 99. CBRHMC SAMPLIB Member (Part 5 of 8)

```
  ELSE                            /* Otherwise there is nothing
                                     special about this object
                                     name (i.e. so overrides
                                     will not be allowed)       */
    SELECT (&STORCLAS)

    WHEN ('DB2DASD')              /* If current storage class
                                     indicates that object is on
                                     DASD                       */
      SET &MGMTCLAS = 'MAG30D'    /* Set management class to
                                     indicate the DASD management
                                     specifications             */
    WHEN ('OLIBRARY')             /* If current storage class
                                     indicates that object is on
                                     optical                    */
      SET &MGMTCLAS = 'OPT6D'     /* Set management class to
                                     indicate the optical
                                     management specifications  */
    OTHERWISE                     /* Otherwise the storage class
                                     is invalid                 */
      EXIT CODE(23)               /* Indicate that an error
                                     occurred                   */
    END
WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT001' | &DSN = 'CLLCT002'
    | &DSN = 'CLLCT003' | &DSN = 'CLLCT004' | &DSN = 'CLLCT005'
    | &DSN = 'CLLCT006' | &DSN = 'CLLCT007' | &DSN = 'CLLCT008'
    | &DSN = 'CLLCT009')          /* For the collections which
                                     do not allow overrides     */
  SELECT (&STORCLAS)

  WHEN ('DB2DASD')                /* If current storage class
                                     indicates that object is on
                                     DASD                       */
    SET &MGMTCLAS = 'MAG30D'      /* Set management class to
                                     indicate the DASD management
                                     specifications             */
  WHEN ('OLIBRARY')               /* If current storage class
                                     indicates that object is on
                                     optical                    */
    SET &MGMTCLAS = 'OPT6D'       /* Set management class to
                                     indicate the optical
                                     management specifications  */
  OTHERWISE                       /* Otherwise the storage class
                                     is invalid                 */
    EXIT CODE(24)                 /* Indicate that an error
                                     occurred                   */
  END
OTHERWISE                         /* Otherwise the collection name
                                     is invalid                 */
  EXIT CODE(25)                   /* Indicate that an error
                                     occurred                   */
END                               /* Select                     */
```

Figure 99. CBRHMC SAMPLIB Member (Part 6 of 8)

```
/*********************************************************************/
/* CHANGE                                                            */
/*********************************************************************/
 IF &ACSENVIR = 'CHANGE' THEN         /* Object is being changed    */
   SELECT (&STORCLAS)

   WHEN ('DB2DASD')                   /* If current storage class
                                         indicates that object is on
                                         DASD                       */
     IF &MGMTCLAS ¬= 'MAGONLY' AND
        &MGMTCLAS ¬= 'MAG30D' THEN    /* If the specified management
                                         class value is not a valid
                                         override                   */
       SET &MGMTCLAS = 'MAG30D'       /* Set management class to
                                         indicate the DASD management
                                         specifications             */
   WHEN ('OLIBRARY')                  /* If current storage class
                                         indicates that object is on
                                         optical                    */
     SET &MGMTCLAS = 'OPT6LIB'        /* Set management class to
                                         indicate the optical
                                         management specifications  */
   WHEN ('TAPESC')                    /* If current storage class
                                         indicates that object is on
                                         tape                       */
     SET &MGMTCLAS = 'OPTTAPE'        /* Set management class to
                                         indicate the tape
                                         management specifications  */
   OTHERWISE                          /* Otherwise the storage class
                                         is invalid                 */
     EXIT CODE(26)                    /* Indicate that an error
                                         occurred                   */
   END
/*********************************************************************/
/* CLASS TRANSITION                                                  */
/*********************************************************************/
 IF &ACSENVIR = 'CTRANS' THEN         /* Object is being processed
                                         as a result of a class
                                         transition                 */
   SELECT (&STORCLAS)

   WHEN ('OLIBRARY')                  /* If current storage class
                                         indicates that object is on
                                         optical                    */
     SELECT (&MGMTCLAS)

     WHEN ('OPT6D')                   /* If current management class
                                         indicates optical management
                                         specifications             */
       SET &MGMTCLAS = 'OPT6LIB'      /* Set management class to
                                         indicate the appropriate
                                         optical management
                                         specifications             */
     WHEN ('MAG30D')                  /* If current management class
                                         indicates DASD management
                                         specifications             */
       SET &MGMTCLAS = 'MAG30LIB'     /* Set management class to
                                         indicate the appropriate
                                         optical management
                                         specifications             */
     OTHERWISE                        /* Otherwise the management
                                         class is invalid           */
       EXIT CODE(27)                  /* Indicate that an error
                                         occurred                   */
     END
```

Figure 99. CBRHMC SAMPLIB Member (Part 7 of 8)

```
   WHEN ('TAPESC')                   /* If current storage class
                                        indicates that object is on
                                        tape                        */
     SELECT (&MGMTCLAS)

     WHEN ('OPT6LIB')                /* If current management class
                                        indicates optical management
                                        specifications              */
       SET &MGMTCLAS = 'OPTTAPE'     /* Set management class to
                                        indicate the appropriate
                                        tape management
                                        specifications              */
     WHEN ('MAG30LIB')               /* If current management class
                                        indicates optical management
                                        specifications              */
       SET &MGMTCLAS = 'TAPSEVEN'    /* Set management class to
                                        indicate the appropriate
                                        tape management
                                        specifications              */
     OTHERWISE                       /* Otherwise the management
                                        class is invalid            */
       EXIT CODE(28)                 /* Indicate that an error
                                        occurred                    */
     END
   OTHERWISE                         /* Otherwise the storage class
                                        is invalid                  */
     EXIT CODE(29)                   /* Indicate that an error
                                        occurred                    */
   END
END
```

Figure 99. CBRHMC SAMPLIB Member (Part 8 of 8)

CBRHSG

SAMPLIB member CBRHSG, as shown in Figure 100 on page 450, provides a storage group ACS routines for OAM objects.

```
/*********************************************************************/
/*                                                                   */
/* $SEG(CBRHSG) COMP(OSMC) PROD(OAM):                                */
/*                                                                   */
/* OAM Sample Storage Group ACS Routine                              */
/*                                                                   */
/* FUNCTION:SUPPLY A STORAGE GROUP FOR OAM OBJECTS                   */
/*                                                                   */
/* OPERATION:Select a storage group based upon the collection name   */
/*           specified                                               */
/*                                                                   */
/* NOTES:    In this implementation, the collection name is used as  */
/*           the basis for determining whether explicit values will */
/*           be considered. This approach and all of the processing */
/*           indicated below is one of many differing possibilities */
/*           and is only for purposes of illustration to demonstrate*/
/*           the types of processing that can be accomplished in an */
/*           ACS routine. Actual implementations will vary.          */
/*                                                                   */
/* ASSUMPTIONS:                                                      */
/*   Collection name format                                          */
/*     CLLCT0mn                                                      */
/*       where m = 0 or 1 (0 indicates that explicit values will be  */
/*                         ignored, 1 indicates that explicit values */
/*                         are considered in some cases and may      */
/*                         result in an override)                    */
/*       where n = 0 - 9                                             */
/*   Valid storage groups:                                           */
/*     SGROUP00 - SGROUP09                                           */
/*                                                                   */
/*********************************************************************/
/*     S T O R A G E   G R O U P   D E F I N I T I O N S             */
/*                                                                   */
/*Relevant                                                           */
/*Fields    SGROUP00 SGROUP01 SGROUP02 SGROUP03 SGROUP04 SGROUP05    */
/*--------- -------- -------- -------- -------- -------- --------    */
/*SG                                                                 */
/*TYPE       OBJECT   OBJECT   OBJECT   OBJECT   OBJECT   OBJECT     */
/*                                                                   */
/*QUALIFIER  GROUP00  GROUP01  GROUP02  GROUP03  GROUP04  GROUP05    */
/*                                                                   */
/*CYCLE                                                              */
/*START        00       01       02       03       04       05       */
/*                                                                   */
/*CYCLE                                                              */
/*END          03       04       05       06       07       08       */
/*                                                                   */
/*LIBRARY                                                            */
/*NAMES       LIB1     LIB1     LIB1     LIB1     LIB1     LIB1      */
/*                              LIB2     LIB3     LIB2     LIB2      */
/*                                                LIB3     LIB3      */
/*                                                                   */
/*VOLUME                                                             */
/*FULL         32       32       32       32       32       32       */
/*                                                                   */
```

Figure 100. CBRHSG SAMPLIB Member (Part 1 of 3)

```
/*DRIVE                                                                */
/*START       099      099      099      099      099      099        */
/*                                                                     */
/*WRITE                                                                */
/*ERROR       YES      YES      YES      YES      YES      YES        */
/*                                                                     */
/*                                                                     */
/*                                                                     */
/*Relevant                                                             */
/*Fields    SGROUP06 SGROUP07 SGROUP08 SGROUP09                        */
/*--------- -------- -------- -------- --------                        */
/*SG                                                                   */
/*TYPE       OBJECT   OBJECT   OBJECT   OBJECT                         */
/*                                                                     */
/*                                                                     */
/*QUALIFIER  GROUP06  GROUP07  GROUP08  GROUP09                        */
/*                                                                     */
/*CYCLE                                                                */
/*START        06       07       08       09                           */
/*                                                                     */
/*CYCLE                                                                */
/*END          09       10       11       12                           */
/*                                                                     */
/*LIBRARY                                                              */
/*NAMES       LIB1     LIB1     LIB1     LIB1                          */
/*            LIB2     LIB2     LIB2     LIB2                          */
/*            LIB3     LIB3     LIB3     LIB3                          */
/*                                                                     */
/*VOLUME                                                               */
/*FULL         32       32       32       32                           */
/*                                                                     */
/*DRIVE                                                                */
/*START       099      099      099      099                           */
/*                                                                     */
/*WRITE                                                                */
/*ERROR       YES      YES      YES      YES                           */
/*                                                                     */
/*********************************************************************/

 PROC STORGRP

 FILTLIST DSN_NAMES INCLUDE(CLLCT0%%)

 IF &DSN = &DSN_NAMES THEN            /* If the first 6 characters of
                                         the collection name are
                                         valid                         */
   IF &DSN ¬= ' ' THEN                /* If the collection name is
                                         not blank (this test will
                                         always pass, but allows for
                                         the apparent assignment of
                                         the 'POOL' storage group
                                         which is a requirement of
                                         a storage group ACS
                                         routine)                      */
```

Figure 100. CBRHSG SAMPLIB Member (Part 2 of 3)

```
/*********************************************************************/
/* Map the collection name to a storage group, where the last digit  */
/* in the collection name corresponds to the last digit of the       */
/* storage group.                                                    */
/*********************************************************************/
     SELECT
     WHEN (&DSN = 'CLLCT000' | &DSN = 'CLLCT010')
       SET &STORGRP = 'SGROUP00'
     WHEN (&DSN = 'CLLCT001' | &DSN = 'CLLCT011')
       SET &STORGRP = 'SGROUP01'
     WHEN (&DSN = 'CLLCT002' | &DSN = 'CLLCT012')
       SET &STORGRP = 'SGROUP02'
     WHEN (&DSN = 'CLLCT003' | &DSN = 'CLLCT013')
       SET &STORGRP = 'SGROUP03'
     WHEN (&DSN = 'CLLCT004' | &DSN = 'CLLCT014')
       SET &STORGRP = 'SGROUP04'
     WHEN (&DSN = 'CLLCT005' | &DSN = 'CLLCT015')
       SET &STORGRP = 'SGROUP05'
     WHEN (&DSN = 'CLLCT006' | &DSN = 'CLLCT016')
       SET &STORGRP = 'SGROUP06'
     WHEN (&DSN = 'CLLCT007' | &DSN = 'CLLCT017')
       SET &STORGRP = 'SGROUP07'
     WHEN (&DSN = 'CLLCT008' | &DSN = 'CLLCT018')
       SET &STORGRP = 'SGROUP08'
     WHEN (&DSN = 'CLLCT009' | &DSN = 'CLLCT019')
       SET &STORGRP = 'SGROUP09'
     OTHERWISE
       EXIT CODE(30)
     END
   ELSE
     DO
       SET &STORGRP = 'POOL'
       EXIT CODE(31)
     END
 ELSE
   EXIT CODE(32)
 END
```

Figure 100. CBRHSG SAMPLIB Member (Part 3 of 3)

Appendix C. Understanding Databases for OAM Diagnosis

OAM uses DB2 databases to store information about objects and to store the objects themselves. This appendix documents diagnosis and modification or tuning information to help diagnose OAM problems. It contains information on DB2 databases and should be used only for diagnosis.

OAM uses the following DB2 databases:

- **Object Storage Databases**—Contain an object directory table and optional object storage tables.
- **Object Administration Database**—Contains the relationship between identifiers and the names of storage classes, management classes, and collections.
- **Optical Configuration Database**—Contains information about the optical hardware configuration and the optical disk volumes.

Object Storage Databases

The object storage databases are a set of DB2 databases containing two types of data: descriptive information about objects and objects stored at the DASD level of the OAM storage hierarchy. Each Object storage group has an object storage database.

Each object storage database contains tables for an object directory and object storage. The object directory table contains descriptive information about each object. Object storage tables contain the objects themselves. There is a separate table space for each table. Each database has three tables:

- An object directory table (contains descriptive information about objects)
- A 4K table (contains small objects)
- A 32K table (contains large objects)

There are multiple object storage databases, each containing the three tables mentioned above, that are supported. Table 45 shows the tables and table space names.

Table 45. Object Storage Database Naming Conventions

Database Name — hlq		
Table Name	**Table Space Name**	**Contents**
hlq.OSM_OBJ_DIR	OSMDTS	Object directory
hlq.OSM_04K_OBJ_TBL	OSMOTS04	Small objects
hlq.OSM_32K_OBJ_TBL	OSMOTS32	Large objects

Sample programs that define these databases and tables are shipped with OAM. These programs must be updated to meet the requirements of your installation before they are run. See “Appendix B. Sample Library Members” on page 381 for the sample jobs CBRIALC0 and CBRISQL0.

The tables defined by the storage administrator will not be used by OAM unless they are related to an Object storage group through the services of ISMF. This relationship results in the definition of the DASD level of the OAM hierarchy (see Figure 101 on page 454) for the specified Object storage group, as well as the object directory for all levels of the object storage hierarchy in that storage group.

The object directory table from each three-table set contains an entry for each object stored in an Object storage group. The object itself may exist in one of the object storage tables on DASD or on optical disk.

Table indexes are necessary for performance. Within an Object storage group, the directory table has three indexes, and each object storage table has a single index. All indexes are unique. Indexes are searched in ascending sequence (ASC).

Note: You must calculate the space required for the indexes separately because it is not included in the directory and object field sizes outlined in each table.

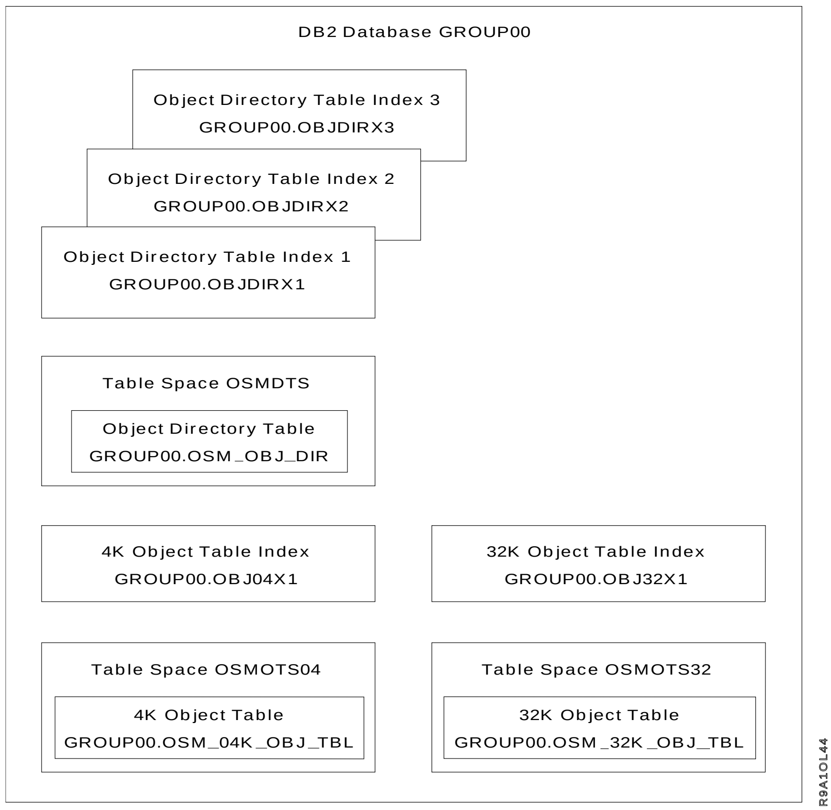

Figure 101. Object Storage Group Database Structure

Object Directory Tables

The object directory tables contain information about objects. OAM keeps track of all objects in the storage hierarchy by recording the collection name identifier, the object name, and other pertinent information in the object directory tables. The object directory tables contain entries for locating and describing objects in the storage hierarchy. OSR creates a directory entry for each object when the object is stored. OSMC uses the directory table to determine which objects need to be processed during each management cycle for an Object storage group.

Table 46 on page 455 shows the fields for an entry in an object directory table.

The object directory table has three indexes:

- **Index 1**—Object Creation Time Stamp
- **Index 2**—Pending Action Date, Collection Name Identifier, and Object Creation Time Stamp

- **Index 3**—Object Name and Collection Name Identifier

Table 46. Object Directory Table

Column Description	Column Name	DB2 Field Type and Data Size	Indexes Where Used
Data Format Version	ODVER	CHAR (1)	
Object Size	ODSIZE	INTEGER	
Object Creation Time Stamp	ODCREATS	TIMESTAMP	Index 1, Index 2
Expiration Date	ODEXPDT	DATE	
Last Referenced Date	ODLREFDT	DATE	
Pending Action Date	ODPENDDT	DATE	Index 2
Management Class Assignment Date	ODMCASDT	DATE	
Storage Class Identifier	ODSCNUM	SMALLINT	
Management Class Identifier	ODMCNUM	SMALLINT	
Object Location Flag	ODLOCFL	CHAR (1)	
Object Active Volume Serial Number	ODLSLOC	CHAR (6)	
Primary Copy Location Token	ODSECLOC	INTEGER	
Object Backup Volume Serial Number	ODBKLOC	CHAR (6)	
Backup Copy Location Token	ODBKSEC	INTEGER	
Second Backup Copy Volume Serial Number	ODBK2LOC	CHAR (6)	
Second Backup Copy Location Token	ODBK2SEC	INTEGER	
Collection Name Identifier	ODCLID	INTEGER	Index 2, Index 3
Object Name	ODNAME	VARCHAR (44)	Index 3

Note:
1. All columns are created with the NOT NULL attribute.
2. All indexes are unique by concatenation of identified columns.
3. All indexes are ordered in ascending value sequence.
4. Index 1 is a cluster index.
5. Maximum size of an object directory entry in bytes: 106.

Table 47 explains the column contents of an object directory table entry.

Table 47. Object Directory Table Field Contents

Column Description	Column Contents
Data Format Version	3
Object Size	Object size in bytes
Object Creation Time Stamp	Compressed form (DB2 format) (yyyy.mm.dd-hh:mm:ss.mmmmmm)

Table 47. Object Directory Table Field Contents (continued)

Column Description	Column Contents
Expiration Date	Compressed form (DB2 format) **0001-01-01** Use MC expiration **yyyy-mm-dd** Explicit expiration date **9999-12-31** Never expire
Last Referenced Date	Compressed form (DB2 format) Set to 0001-01-01 upon creation
Pending Action Date	Compressed form (DB2 format)
Management Class Assignment Date	Compressed form (DB2 format) Set to creation date on creation; otherwise, last date MC changed
Storage Class Identifier	Number identifying this storage class (associated with storage class name via Storage Class Identifier Table)
Management Class Identifier	Number identifying this management class (associated with management class name via Management Class Identifier Table)
Object Location Flag	Blank—Optical copy T—Tape copy D—DASD copy
Object Active Volume Serial Number	Standard MVS volume serial number (or blanks)
Primary Copy Location Token	If optical volume: Token for relative sector location (or zeros) of VTOC entry. If tape volume: Tape blockid
Object Backup Volume Serial Number	Standard MVS volume serial number (or blanks)
Backup Copy Location Token	If optical volume: Token for relative sector location (or zeros) of VTOC entry. If tape volume: Tape blockid
Second Backup Copy Location Token	Volume serial number for the optical or tape volume that contains the second backup copy of the object in the corresponding row in the table.
Second Backup Copy Sector or Block ID	The optical volume sector location or the tape volume block ID on the volume in the ODBK2LOC field where the second backup copy of the object in the corresponding row in the table resides.
Collection Name Identifier	Number identifying the collection name (associated with collection name via Collection Name Identifier Table)
Object Name	Standard MVS data set name

Object Storage Tables

The object storage tables provide DASD storage for objects. Objects are stored in the 4K or 32K table, depending on size. If an object is 3980 bytes or smaller, it is stored in the 4K table. If the object is larger than 3980 bytes, it is stored in the 32K table.

Objects stored in the 32K table may be broken into segments and stored as rows. Rows in the 32K table can contain up to 32 640 bytes of object data.

Table 48 shows the contents of an entry in an object storage table.

Each object storage table has one index. The 4K table index is the concatenation of the collection name ID and object name in ascending-order sequence. The 32K table index is the concatenation of the collection name ID, object name and segment number in ascending-order sequence. When objects are retrieved, they are ordered by object segment number.

Table 48. Object Storage Table

Column Description	Column Name	DB2 Field Type and Data Size	Indexes Where Used
Data Format Version	OTVER	CHAR (1)	
Segment Number	OTSEG	SMALLINT	Index 1
Collection Name Identifier	OTCLID	INTEGER	Index 1
Object Name	OTNAME	VARCHAR (44)	Index 1
Object Data Segment	OTOBJ	LONG VARCHAR	

Note:
1. All columns are created with the NOT NULL attribute.
2. The object table columns are the same for the 4K and 32K tables.
3. Segment number is *not* used in the 4K table.
4. The index on each table is a unique cluster index.
5. Maximum sizes of object table entries:
   4K Table 3 980
   32K Table 32 640

Object Administration Database

Each object stored in the OAM storage hierarchy is part of a collection and is assigned a storage class and management class. These assignments are recorded in the object's directory entry. To conserve DASD space, OAM stores an identifier that represents those names instead of recording the names in each directory entry. OAM requires two tables to relate the identifiers to the actual storage class and management class names, and a third table to describe collections (see Figure 102 on page 458). Table 49 on page 458 through Table 57 on page 469 are used for diagnostic reference.

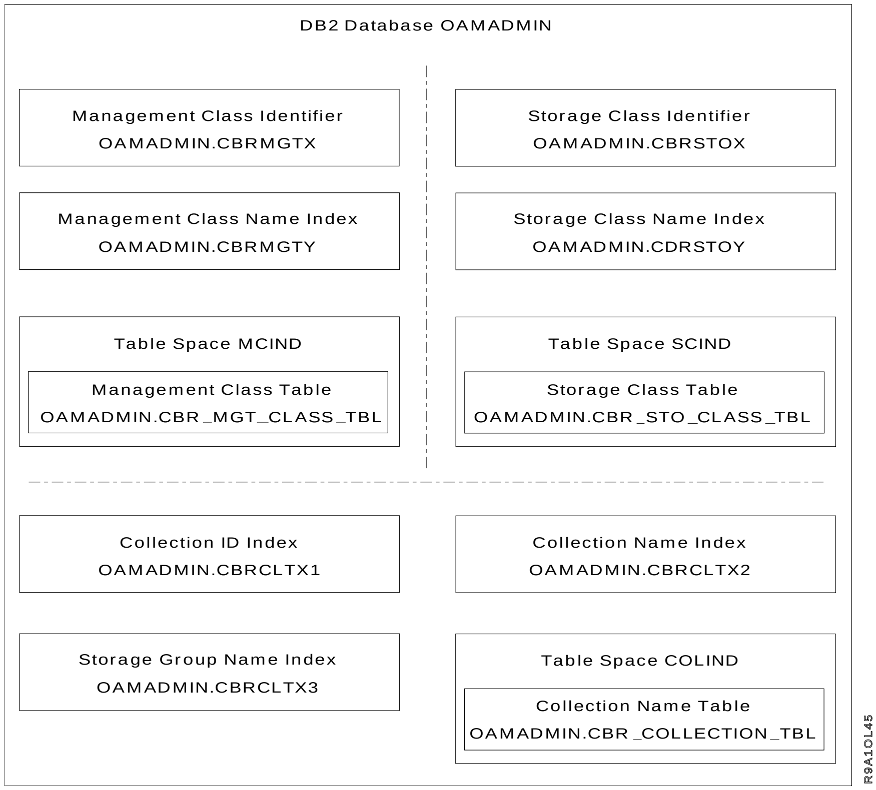

Figure 102. Object Administration Database Structure

Storage Class Identifier Table

Each object stored in the OAM storage hierarchy is assigned a storage class. This assignment is recorded in the object's directory entry as a storage class identifier. The storage class identifier table maps the identifier to the actual storage class name (see Table 49).

The storage class identifier table has one index: a unique, ascending cluster index on the storage class identifier.

Table 49. Storage Class Identifier Table

Column Description	Column Name	DB2 Field Type and Data Size	Max Byte	Index Definition
Storage Class Identifier	ODSCNUM	SMALLINT	(2)	Unique ASC
Storage Class Name	ODSCNAME	VARCHAR (30)	(32)	Unique ASC
Index Structure: ASC Ascending-sequence				

Management Class Identifier Table

Each object stored in the OAM storage hierarchy is assigned a management class. This assignment is recorded in the object's directory entry as a management class identifier. The management class identifier table maps the identifier to the actual management class name (see Table 50 on page 459).

The management class identifier table has one index: a unique, ascending cluster index on the management class identifier.

Table 50. Management Class Identifier Table

Column Description	Column Name	DB2 Field Type and Data Size	Max Byte	Index Definition
Management Class Identifier	ODMCNUM	SMALLINT	(2)	Unique ASC
Management Class Name	ODMCNAME	VARCHAR (30)	(32)	Unique ASC
Index Structure: ASC Ascending-sequence				

Collection Name Identifier Table

Each object stored in the OAM storage hierarchy is a member of a collection. The name of the collection to which an object belongs is recorded in the object's directory entry as a collection name identifier. The collection name identifier table maps the identifier to the actual collection name (see Table 51). In addition, the collection name identifier table contains information about the storage class and management class for the collection and the name of the storage group that contains all members of the collection (see Table 52).

Table 51. Collection Name Identifier Table

Column Description	Column Name	DB2 Field Type and Data Size	Max Byte	Index Definition
Storage Class Name	ODCLSCNM	VARCHAR (30)	(32)	
Management Class Name	ODCLMCNM	VARCHAR (30)	(32)	
Directory Token (*)	ODCLSGNM	VARCHAR (30)	(32)	ASC
Collection Name Identifier (*)	ODCLID	INTEGER (4)	(4)	Unique ASC Cluster
Collection Name (*)	ODCLNAME	VARCHAR (44)	(46)	Unique ASC
Total Bytes per Table Entry	138	146		
Index Structure: (*) Indexed fields ASC Ascending-sequence				

Table 52. Collection Name Identifier Table Contents

Column Description	Contents
Storage Class Name	Default initial storage class for all objects in this collection (can be overridden by explicit storage class on OSREQ STORE)
Management Class Name	Default initial management class for all objects in this collection (can be overridden by explicit management class on OSREQ STORE)
Directory Token	Storage group name
Collection Name Identifier	Numeric index identifying the collection that includes this object (used to improve DASD space usage in object tables)
Collection Name	Standard MVS data set name

Optical Configuration Database

The optical configuration database (CBROAM) defines the optical hardware configuration and all of the optical volumes. It is a DB2 database and consists of the following tables:

- Library table (OLIBRARY)
- Drive table (DRIVE)
- Slot table (SLOT)
- Volume table (VOLUME)
- Deleted-Objects table (DELOBJT)
- Tape Volume table (TAPEVOL).

Figure 103 shows the organization of the optical configuration database. There are six table spaces, each containing a different table and its associated indexes.

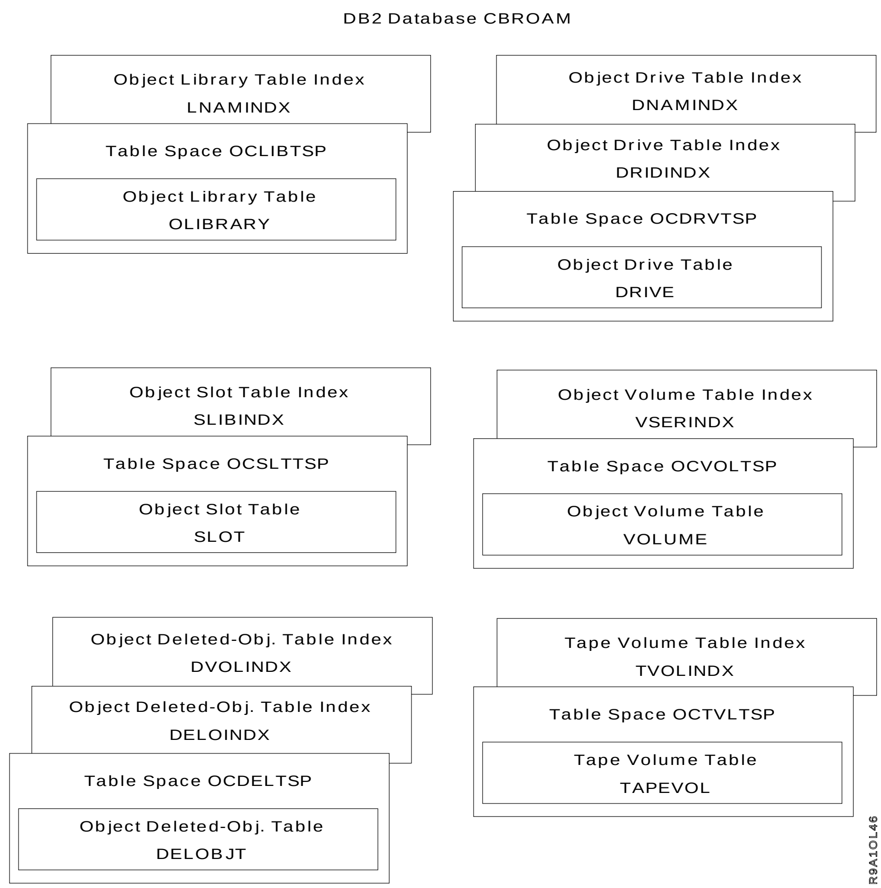

Figure 103. Optical Configuration Database

Table 53 on page 461, Table 54 on page 463, Table 55 on page 464, Table 56 on page 465, Table 57 on page 469, and Table 58 on page 470 describe the six tables in the optical configuration database. A description of special characteristics follows each table.

Each table is in its own table space and each table has at least one index.

Note: If DB2 searches a table without using an index, it must search the entire table space. Each table is within its own table space to decrease search time when an index is not used.

The deleted-objects table is used only with OAM rewritable support.

The following explains the information in the DB2 Attribute column of each figure:

NN This column is declared with the NOT NULL attribute. A value must be supplied.

NND This column is declared with the NOT NULL WITH DEFAULT attribute. If some other value is not given, the following DB2 data types and default values are supplied:

CHARACTER blanks
SMALLINT zero
INTEGER zero
DATE current date

The value column of each figure shows the columns that require specific values. When the value column contains any information other than blank, that column must contain a specific value. The possible values are listed in the value column. The explanations about each table define the specific values.

Library Table

The library table contains one row for each optical library. The DB2 name of the library table is OLIBRARY. The table is defined in table space OCLIBTSP. Table 53 describes the attributes of the columns in the OLIBRARY table. A row is inserted into the library table the first time that an optical library with a given name is defined using the ISMF Library Management application.

Table 53. Library Table Column Description

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Optical library name	NAME	U	CHAR(8)	NN		NAME
Online status	ONLINE		CHAR(1)	NN	Y|N	ONLINE
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Path status	PATHSTAT**		CHAR(1)	NN	P|A	CURRENT_PATH
Current command	COMMAND		CHAR(5)	NND		CURRENT_COMMAND
Primary CTC device number	PRIMCTC		CHAR(4)	NND		PRIMARY_CTC
Primary port	PRIMPORT**		CHAR(1)	NND		PRIMARY_PORT
Alternate CTC device number	ALTCTC		CHAR(4)	NND		ALTERNATE_CTC
Alternate port	ALTPORT**		CHAR(1)	NND		ALTERNATE_PORT
Fault code	FAULT**		CHAR(3)	NND		FAULT_CODE
Library type	OLIBTYPE		CHAR(1)	NN	P|R	LIBRARY_TYPE
Number of slots	NUMSLOTS		INTEGER	NND		SLOTS
Number of empty slots	NUMESLOT		INTEGER	NND		EMPTY_SLOTS
Number of drives	NUMDRVS		SMALLINT	NN		DRIVES
Current recovery command	RCOMMAND**		CHAR(5)	NND		RECOVERY_COMMAND

Table 53. Library Table Column Description (continued)

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Device type	DEVTYPE		CHAR(8)	NND		DEVICE_TYPE
Library description	LIBRDES		CHAR(120)	NND		LIBRARY_ DESCRIPTION
Controlling Library	CLIBRARY		CHAR(8)	NND		CONTROLLING_LIBRARY
Default media type	MEDIATYP		CHAR(8)	NND		DEFAULT_MEDIA_TYPE
Library index	LIBINDEX		SMALLINT	NND		LIBRARY_INDEX
OAM XCF member name	MEMBER		CHAR(16)	NND		OAM_XCF_MEMBER
Default pseudo library	PLIBRARY		CHAR(8)	NND		DEFAULT_PSEUDO_LIBRARY
Note: ** Applies only to 9246/9247.						

There is a unique index on the NAME column; therefore, each optical library defined in the library table must have a unique name.

The online status, operational status, path status, library type, and device type columns in the library table require specific values:

ONLINE
Indicates the online status of the library to a specific system.
- **Y** Library is online to a system.
- **N** Library is offline to a system.

OPERATNL
Indicates the operational status of the library.
- **Y** Library is operational.
- **N** Library is not operational.

PATHSTAT
Indicates which path is being used.
- **P** Primary path is being used.
- **A** Alternate path is being used.

Note: If this is a pseudo library, the PATHSTAT column is blank. Path status does not apply to pseudo libraries.

OLIBTYPE
Indicates the library type.
- **P** This is a pseudo library.
- **R** This is a real library.

DEVTYPE
Indicates the device type associated with the library.
- **3995-111** This is a 3995 Model 111.
- **3995-112** This is a 3995 Model 112.
- **3995-113** This is a 3995 Model 113.
- **3995-131** This is a 3995 Model 131.
- **3995-132** This is a 3995 Model 132.
- **3995-133** This is a 3995 Model 133.
- **3995-SW3** This is a 3995 Model SW3. Valid for pseudo libraries only.
- **3995-SW4** This is a 3995 Model SW4. Valid for pseudo libraries only.
- **3995-C3A** This is a 3995 Model C3A.
- **3995-C12** This is a 3995 Model C12.
- **3995-C16** This is a 3995 Model C16.
- **3995-C18** This is a 3995 Model C18.
- **3995-C32** This is a 3995 Model C32.

3995-C34	This is a 3995 Model C34.
3995-C36	This is a 3995 Model C36.
3995-C38	This is a 3995 Model C38.
9246	This is a 9246.

MEMBER The instance of OAM, to which the library is online, that is currently controlling and managing the library within the parallel sysplex.

PLIBRARY This is either the name of the default target pseudo library for volumes ejected from this library if the library is a real library type, or this field is blank.

Drive Table

The drive table contains one row for each optical drive, whether stand-alone or library-resident. The DB2 name of the drive table is DRIVE. The table is defined in table space OCDRVTSP. Table 54 describes the attributes of the columns in the drive table. A row is inserted into the drive table the first time that an optical disk drive with a given name is defined using the ISMF Library Management application.

Table 54. Drive Table Column Description

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Optical drive name	NAME	U	CHAR(8)	NN		NAME
Optical library name	OLIBRARY		CHAR(8)	NN		OLIBRARY
CTC device number	CTC	P	CHAR(4)	NN		CTC
SCSI bus address	SCSI**	P	CHAR(1)	NN		SCSI
Logical unit number	LUN**	P	CHAR(1)	NN		LUN
Online status	ONLINE		CHAR(1)	NN	Y|N	ONLINE
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Library drive number	LDRIVENO		CHAR(1)	NND		DRIVE_NUMBER
Drive type	DRIVTYPE		CHAR(1)	NN	L|S	DRIVE_TYPE
Device type	DEVTYPE		CHAR(8)	NND		DEVICE_TYPE
Drive description	DRIVDES		CHAR(120)	NND		DRIVE_DESCRIPTION
Physical Drive Number	DRIVENUM		SMALLINT	NND		PHY_DRIVE_NUMBER
OAM XCF member name	MEMBER		CHAR(16)	NND		OAM_XCF_MEMBER
Note: **Applies only to 9247.						

There is a unique index on the NAME column; therefore, each optical disk drive defined in the drive table must have a unique name.

The combination of CTC, SCSI, and LUN must be unique for each optical drive defined in the drive table, because there is a partitioned index on the CTC, SCSI, and LUN columns. CTC, SCSI, and LUN constitute the device address of the optical disk drive. This address is used by OAM to address the optical drive during I/O operations.

The online status, operational status, drive type, and device type columns in the drive table require specific values:

ONLINE Indicates whether the optical drive is online or offline to a particular system.

- **Y** The optical drive is online to a specific system.
- **N** The optical drive is offline to a specific system.

OPERATNL Indicates the operational status of the optical drive.

- **Y** Optical drive is operational.
- **N** Optical drive is not operational.

DRIVTYPE Indicates the type of optical drive.

- **L** Optical drive is in a library.
- **S** Optical drive is a stand-alone.

DEVTYPE Indicates the device type of the optical disk drive.

- **9247** This is a 9247.
- **3995-131** This is a 3995 Model 131.
- **3995-132** This is a 3995 Model 132.
- **3995-133** This is a 3995 Model 133.
- **3995SW3** This is a 3995 Model SW3.
- **3995SW4** This is a 3995 Model SW4.

MEMBER The instance of OAM, to which the drive is online, that is currently controlling and managing the optical drive within the parallel sysplex.

Slot Table

The slot table contains one row for each slot in an optical library. The DB2 name of the slot table is SLOT. The table is defined in table space OCSLTTSP. The row for each slot gives the status of the slot. Also, there is a row for the optical library input/output station and the optical library cartridge access mechanism. Table 55 describes the attributes of the columns in the SLOT table.

Table 55. Slot Table Column Description

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Slot name	NAME	P	CHAR(3)	NN		NAME
Optical library name	OLIBRARY	P	CHAR(8)	NN		OLIBRARY
Occupied status	OCCUPIED		CHAR(1)	NN	Y|N	OCCUPIED
Operational status	OPERATNL		CHAR(1)	NN	Y|N	OPERATIONAL
Side 0 volume serial number	VOLSER0		CHAR(6)	NND		VOLSER0
Side 1 volume serial number	VOLSER1		CHAR(6)	NND		VOLSER1
Note: This table is not used for 3995 support.						

The combination of NAME and OLIBRARY must be unique for each slot defined in the slot table because there is a partitioned index on the NAME and OLIBRARY columns.

During OAM initialization, all necessary rows are inserted into the slot table based on the optical libraries defined in the library table.

The occupied status and operational status columns in the slot table require specific values:

OCCUPIED
Indicates the status of the slot within the library.
Y Slot is occupied.
N Slot is not occupied.

OPERATNL
Indicates the operational status of the slot within the library.
Y Slot is operational.
N Slot is not operational.

Volume Table

The volume table contains one row for each optical disk volume. The DB2 name of the volume table is VOLUME. The table is defined in table space OCVOLTSP. Table 56 describes the attributes of the columns in the volume table. Two rows are inserted into the volume table when the two optical volumes comprising an optical disk are identified to OAM. The two optical volumes are identified to OAM when the following conditions exist:

- The two volumes are labeled on a stand-alone optical disk drive in response to an OAM LABEL command.
- The two volumes are labeled on a library-resident optical disk drive when the operator enters an unlabeled optical cartridge into the input/output station of an optical library.
- The two volumes comprising an already labeled, but unknown, optical cartridge are verified as part of the cartridge being entered into an optical library.

Should OAM discover minor discrepancies with the Volume Table at initialization, the following recovery actions are automatically invoked to circumvent failure of the initialization:

- The row that is in error is skipped over, and a corresponding volume or tape volume control block is not built. A message is issued indicating the row that was skipped and the reason it was bypassed.
- The table row is corrected when a valid value is easily recognizable, and a message is issued stating the correction that is made by OAM and what steps can be taken if the correction is not acceptable to the customer.
- More detailed messages containing recovery actions are provided and issued during OAM initialization for database discrepancies.

Table 56. Volume Table Column Description

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Volume serial number	VOLSER	U	CHAR(6)	NN		VOLSER
Other side VOLSER	OVOLSER		CHAR(6)	NN		OTHER_VOLSER
Location	LOCATION		CHAR(1)	NN	L|S	LOCATION
Slot name	SLOT**		CHAR(3)	NN		SLOT
Library name	OLIBRARY		CHAR(8)	NN		OLIBRARY
Shelf location	SHELFLOC		CHAR(32)	NND		SHELF_LOCATION
Last-mounted date	MNTDATE		DATE	NND		DATE_LAST_MOUNTED
Last-written date	WRTDATE		DATE	NND		DATE_LAST_WRITTEN
Expiration date	EXPDATE		DATE	NND		EXPIRATION_DATE
Eject/Enter date	EJECTDAT		DATE	NND		EJECT/ENTER_DATE

Table 56. Volume Table Column Description (continued)

Column Description	Name	Index	DB2 Type	DB2 Attribute	Value	Report Label
Address of last data sector	LASTDATA**		INTEGER	NN		LAST_DATA_SECTOR
Address of last logical OVTOC sector	LASTVTCL**		INTEGER	NN		LAST_LOGICAL_VTOC_SECTOR
Address of last physical OVTOC sector	LASTVTCP**		INTEGER	NN		LAST_PHYSICAL_VTOC_SECTOR
Storage group name	VOLUMSET		CHAR(8)	NN		STORAGE_GROUP
Volume type	TYPE		CHAR(1)	NN	B|G|S	TYPE
Orientation	ORIENT**		CHAR(1)	NN	0|1	ORIENTATION
Full status	FULL		CHAR(1)	NN	Y|N	FULL
Readable	READABLE		CHAR(1)	NN	Y|N	VOLUME_READABLE_STATUS
Writable	WRITABLE		CHAR(1)	NN	Y|N	VOLUME_WRITABLE_STATUS
Write-protected status	WRTPROT		CHAR(1)	NN	Y|N	WRITE_PROTECTED
Owner information part	OWNERP		CHAR(1)	NND	1|2	OWNER_INFORMATION_POSITION
Owner information	OWNER		CHAR(32)	NND		OWNER_INFORMATION
Free space	FRESPACE		INTEGER	NND		FREE_SPACE
Deleted space	DELSPACE*		INTEGER	NND		DELETED_SPACE
Number of deleted objects	DELCOUNT*		INTEGER	NND		DELETED_OBJECTS
Fragmentation index	FRAGIDX*		SMALLINT	NND		FRAGMENTATION_INDEX
Media type	MEDIATYP		CHAR(2)	NND		MEDIA_TYPE
Volume creation date	CREDATE		DATE	NND		CREATE_DATE
Volume error status	ERRSTAT***		SMALLINT	NND		VOLUME_ERROR_STATUS
Volume empty	VOLEMPTY*		CHAR(1)		Y|N	VOLUME_EMPTY
Deleted objects recount	RECOUNT*		SMALLINT		0|1	DELETED_OBJECTS_ RECOUNT
Volume Capacity	CAPACITY		INTEGER	NND		CAPACITY
OAM XCF member name	MEMBER		CHAR(16)	NND		OAM_XCF_MEMBER
Pseudo library name	PLIBRARY		CHAR(8)	NND		PSEUDO_LIBRARY_FOR_VOLUME
Backup type	BKTYPE		CHAR(6)	NN	B	BKTYPE

Note:

- * DELSPACE, DELCOUNT, FRAGIDX, VOLEMPTY, and RECOUNT apply only to OAM rewritable support.
- **Applies only to 9246/9247.
- ***Applies only to the 3995.

There is a unique index on the VOLSER column; therefore, each optical disk volume defined in the volume table must have a unique volume serial number. The optical disk volume serial number must be unique across all types of media used by the installation. The optical disk volume serial number must not conflict with the volume serial number of a tape volume being used by OAM. The optical disk volume serial number must not conflict with the serial number of any SMS-managed DASD volume or any mounted non-SMS-managed DASD volume.

The columns labeled volume location, volume type, volume orientation, volume full, volume readable, volume writable, write-protected status, owner information position, media type, volume error status, volume empty, and deleted objects recount in the volume table require specific values:

VOLSER The volume serial number on one side of the optical disk.

OVOLSER The volume serial number on the opposite side of the optical disk.

LOCATION The location of the optical volume
- **L** This volume is in a library.
- **S** This volume is on the shelf.

SLOT The 9246 library slot location for the optical volume.

OLIBRARY The library name in which the volume resides.

SHELFLOC The shelf location of the shelf-resident optical volume.

MNTDATE The date that OAM last mounted the volume.

WRTDATE The date that OAM last wrote to the volume.

EXPDATE The expiration date of the volume. This is the date of the latest expiration date of all the objects that reside on the optical volume.

EJECTDAT This is the date the volume was last entered into or ejected out of the optical library.

LASTDATA The address of the last data sector on the optical volume. This field only applies to 9247 volumes.

LASTVTCL The address of the last logical optical VTOC sector on the optical volume. This field only applies to 9247 volumes.

LASTVTCP The address of the last physical optical VTOC sector on the optical volume. This field only applies to 9247 volumes.

VOLUMESET The name of the storage group to which the optical volume is associated.

TYPE The type of the optical volume.
- **B** This is a BACKUP volume.
- **G** This is a GROUPED volume.
- **S** This is a SCRATCH volume.

ORIENT The orientation position of the 9247 optical volume.
- **0** This 9247 volume is stored in orientation 0 in the specified library slot.
- **1** This 9247 volume is stored in orientation 1 in the specified library slot.

  This is always blank for 3995.

FULL The capacity of the optical volume.
- **Y** This volume is full.
- **N** This volume is not full.

READABLE Specifies whether the optical volume is readable.
- **Y** This volume can be read.

N This volume cannot be read.

WRITABLE Specifies whether the optical volume is writable.
- **Y** This volume can be written on.
- **N** This volume cannot be written on.

WRTPROT Specifies whether the optical volume is write-protected.
- **Y** This volume is write-protected.
- **N** This volume is not write-protected.

OWNERP Position of the owner information.
- **1** This is part 1 of the owner information.
- **2** This is part 2 of the owner information.

OWNER The volume owner information.

FRESPACE The available free space left for writing data, in kilobyte units (1 kilobyte = 1 024 bytes) on the optical volume.

DELSPACE The amount of deleted space on a rewritable optical disk.

DELCOUNT The amount of deleted objects marked for deletion from the rewritable optical disk.

FRAGDIX The fragmentation index on a rewritable optical disk.

MEDIATYP The media type of the optical volume.
- **00** This is a 9247 12-inch volume.
- **01** This is a 3995 5.25-inch, single-density, rewritable volume.
- **03** This is a 3995 5.25-inch, single-density, WORM volume.
- **11** This is a 3995 5.25-inch, double-density, rewritable volume.
- **13** This is a 3995 5.25-inch, double-density, WORM volume.
- **15** This is a 3995 5.25-inch, double-density, CCW volume.
- **21** This is a 3995 5.25-inch, quad-density, rewritable volume.
- **23** This is a 3995 5.25-inch, quad-density, WORM volume.
- **25** This is a 3995 5.25-inch, quad-density, CCW volume.
- **31** This is a 3995 5.25-inch, 8x-density, rewritable volume.
- **33** This is a 3995 5.25-inch, 8x-density, WORM volume.
- **35** This is a 3995 5.25-inch, 8x-density, CCW volume.

Note: CCW = continuous composite WORM media. WORM = write-once-read-many media.

CREDATE The date the optical volume was created.

ERRSTAT The error status of the optical volume.
- **0** No error status. This is the initial setting.
- **101** This volume has an entry in the OCDB but AUDIT found no corresponding entry in the outboard inventory.
- **102** The cartridge is missing from its assigned slot in the library (empty slot found).
- **103** AUDIT found the wrong volser in the slot.
- **105** An error occurred when attempting to read the volume serial number while auditing a volume.
- **201** This volume has an entry in the OCDB but REMAP found no corresponding entry in the outboard inventory.

VOLEMPTY Specifies whether the optical volume can be erased.
- **Y** This 3995 rewritable volume can be reformatted.
- **N** This 9247 or 3995 write-once-read-many volume cannot be erased.

RECOUNT Specifies whether a recount of the logically deleted objects, or a summing up of the available deleted kilobytes, is performed.

0 No recount of the number of logically deleted objects, or summing up of the available kilobytes that have been deleted, is performed.

1 A recount of the number of logically deleted objects, or summing up of the available kilobytes that have been deleted, is performed.

MEMBER Specifies the name of the OAM within a parallel sysplex that is currently managing and controlling this optical volume.

- For library-resident optical volumes, this member name is the equivalent of the member name for the library in which this volume currently resides if the library is online. This MEMBER field is blank if the library is offline.
- For shelf-resident optical volumes that are currently mounted on operator-accessible drives, this member name is the equivalent of the member name for the operator-accessible drive (where the operator-accessible drive is online).
- For shelf-resident optical volumes that are not currently mounted on an operator-accessible drive, this member name is blank.

PLIBRARY The name of the pseudo library the volume is assigned to when it is no longer a library-resident volume.

- For shelf-resident optical volumes, this field value is the same as the OLIBRARY column in the DB2 row that represents the volume.
- For library-resident optical volumes the field value is either of the following values:
  - The pseudo library that the volume was associated with, if it was shelf-resident prior to being entered into the library
  - Blank if the volume was not shelf-resident prior to being entered into the library.

| **BKTYPE** This indicates whether this volume is used for first or second
| backup copies of an object when the volume has a type of “B”,
| which indicates that it is a backup volume belonging to an Object
| Backup storage group.

Deleted Objects Table

The deleted objects table contains one row for each object to be deleted. The name of the deleted objects table is DELOBJT. The table is defined in table space OCDELTSP. Table 57 describes the attributes of the columns in the deleted objects table.

The combination of the COLNAME, OBJNAME, VOLSER, and VTOCTOKN columns must be unique throughout the table. However, multiple entries in the table may have the same VOLSER number.

Table 57. Deleted Object Table Column Description

Column Description	Name	Index	DB2 Type	DB2 Attribute	Report Label
Collection name	COLNAME	P	CHAR(44)	NN	COLLECTION_NAME
Object name	OBJNAME	P	CHAR(44)	NN	OBJECT_NAME

Table 57. Deleted Object Table Column Description (continued)

Column Description	Name	Index	DB2 Type	DB2 Attribute	Report Label
Volume serial number	VOLSER	PN	CHAR(6)	NN	VOLSER
VTOC token	VTOCTOKN	P	INTEGER	NN	VTOC_TOKEN
Object size	OBJSIZE		INTEGER	NN	OBJECT_SIZE
Note: The deleted-objects table is used with OAM rewritable support.					

Tape Volume Table

The tape volume table contains one row for each tape volume used by OAM. The DB2 name of the tape volume table is TAPEVOL. The table is defined in table space ODTVLTSP. Table 58 describes the attributes of the columns in the ODTVLTSP table. A row is inserted into the tape volume table for each tape volume used by OAM to track its status.

Note: Should OAM discover minor discrepancies with the tape volume table at initialization, the following recovery actions are automatically invoked to circumvent failure of the initialization:

- The row that is in error is skipped over, and a corresponding volume or tape volume control block is not built. A message is issued indicating the row that was skipped and the reason it was bypassed.
- The table row is corrected when a valid value is easily recognizable, and a message is issued stating the correction that is made by OAM and what steps can be taken if the correction is not acceptable to the customer.
- More detailed messages containing recovery actions are provided and issued during OAM initialization for database discrepancies.

Table 58. Tape Volume Table Column Description

Column Description	Name	DB2 Type	DB2 Attribute	Report Label
Volume Serial number	VOLSER	CHAR(6)	Not Null	VOLSER
Unit Name	UNITNAME	CHAR(8)	Not Null	UNIT_NAME
Media type	MEDIATYP	CHAR(2)	Not Null	MEDIA_TYPE
Storage group name	STORGRP	CHAR(8)	Not Null	STORAGE_GROUP
Volume type	TYPE	CHAR(1)	Not Null	TYPE
Volume creation date	CREDATE	DATE	Not Null	CREATION_DATE
Last mounted date	MNTDATE	DATE	Not Null	DATE_LAST_MOUNTED
Last written date	WRTDATE	DATE	Not Null	DATE_LAST_WRITTEN
Expiration date	EXPDATE	DATE	Not Null	EXPIRATION_DATE
Capacity of tape	CAPACITY	INTEGER	Not Null	CAPACITY
Free space remaining	FRESPACE	INTEGER	Not Null	FREE_SPACE
Block id of last data block written	LSTBLKID	INTEGER	Not Null	LAST_BLOCKID
Percent Full	PFULL	SMALLINT	Not Null	PERCENT_FULL
Number of logical blocks written	NUMLBLKS	INTEGER	Not Null	LOGICAL_BLOCKS_WRITTEN
Number of logical kilobytes written	NUMLKBW	INTEGER	Not Null	LOGICAL_KILOBYTES_WRITTEN

Table 58. Tape Volume Table Column Description (continued)

Column Description	Name	DB2 Type	DB2 Attribute	Report Label
Number of physical kilobytes written	NUMPKBW	INTEGER	Not Null	PHYSICAL_KILOBYTES_WRITTEN
Full status	FULL	CHAR(1)	Not Null	FULL
Readable	READABLE	CHAR(1)	Not Null	VOLUME_READABLE_STATUS
Writable	WRITABLE	CHAR(1)	Not Null	VOLUME_WRITABLE_STATUS
In use status	INUSE	CHAR(1)	Not Null	IN_USE
Copied status	COPIED	CHAR(1)	Not Null	COPIED
Alternate volume	AVOLSER	CHAR(6)	Not Null	ALTERNATE_VOLUME
Number of logical kilobytes deleted	NUMLKBDE	INTEGER	Not Null with Default	LOGICAL_KILOBYTES_DELETED
Tape compaction indicator	COMPACT	CHAR(1)	Not Null with Default	TAPE_COMPACTION_INDICATOR
OAM XCF member name	MEMBER	CHAR(16)	Not Null with Default	OAM_XCF_MEMBER
Physical Identifier	EPI	SMALLINT	Not Null with Default	EPI
Backup type	TYPE	CHAR(1)	Not Null	BKTYPE

There is a unique index on the **VOLSER** column; therefore, each tape volume used by OAM must have a unique volume serial number. The tape volume serial number must be unique across all types of media used by the installation. The tape volume serial number must not conflict with the volume serial number of an optical volume being used by OAM. The tape volume serial number must not conflict with the serial number of any SMS-managed DASD volume or any mounted non-SMS-managed DASD volume.

Rows are dynamically inserted into the TAPEVOL table as unknown scratch tape volumes mounted in a response to a mount scratch request during allocation.

The following describes the columns in the tape volume table:

VOLSER
The volume serial number of the tape volume. All other columns in the tape volume table row apply to this volume.

UNITNAME
The MVS unit name used when the tape volume is initially mounted for OAM use. This unit name is used by OAM whenever this tape volume is subsequently allocated by OAM. This parameter is only valid for stand-alone tape drives. If the tape volume is library-resident, or if an automated or manual tape library dataserver is chosen for the request at the time of allocation, this parameter is ignored (in the case of a library-resident volume mount request), or overridden (in the case of an ATLDS or MTLDS being chosen to handle the request at allocation).

MEDIATYPE
The media type of the tape volume

02 IBM Cartridge System Tape
04 IBM Enhanced Capacity Cartridge System Tape
05 IBM High Performance Cartridge Tape
06 Extended High Performance Cartridge Tape

STORGRP
The name of the Object or Object Backup storage group to which the tape volume is associated.

TYPE
The type of tape volume:

B Backup volume associated with an Object Backup storage group.

G Group volume associated with an Object storage group.

S Scratch volume that can be assigned to either an Object or Object Backup storage group when another volume is needed by OAM.

CREDATE The date that the volume was first used by OAM and when the row for this volume was created in the TAPEVOL table.

MNTDATE The date that the volume was last mounted by OAM.

WRTDATE The date that the volume was last written by OAM.

EXPDATE The expiration date of the volume. The expiration date of the volume is the latest expiration date of all objects that reside on the volume.

CAPACITY The approximate number of kilobytes of data which can be written for the volume allowing variance for different manufactures.

218 554
Represents the approximate number of kilobytes of data that can be written for an IBM Cartridge System Tape written in 18-track format on an IBM 3480 or 3490 Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

437 109
Represents the approximate number of kilobytes of data that can be written for an IBM Cartridge System Tape written in 36-track format on an IBM 3490E Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

874 218
Represents the approximate number of kilobytes of data that can be written for an IBM Enhanced Capacity Cartridge System Tape written in 36-track format on an IBM 3490E Magnetic Tape Subsystem. The installation can overwrite this default capacity by specifying a value between 1 and 2 147 483 646 kilobytes using the TAPECAPACITY parameter of the SETOAM command.

9 764 864
Represents the approximate number of kilobytes of data for an IBM High Performance Cartridge Tape written in 128-track format on an IBM TotalStorage Enterprise High Performance Tape System 3590 Model B subsystem.

If the IBM TotalStorage Enterprise High Performance Tape System 3590 Model B subsystem is installed in native non-emulation mode, this value is returned from the drive and is used as an approximation that is close to the actual value.

19 530 752
Represents the approximate number of kilobytes of data for an IBM Extended High Performance Cartridge Tape written in 128-track format on an IBM TotalStorage Enterprise High Performance Tape System 3590 Model B subsystem.

If the IBM TotalStorage Enterprise High Performance Tape System 3590 Model B subsystem is installed in native non-emulation mode, this value is returned from the drive and is used as an approximation that is close to the actual value.

19 530 752
: Represents the approximate number of kilobytes of data for an IBM High Performance Cartridge tape written in 256-track recording technology on an IBM TotalStorage Enterprise High Performance Tape System 3590 Model E subsystem.

  This value is returned from the drive and is used as an approximation that is close to the actual value.

39 061 504
: Represents the approximate number of kilobytes of data for an IBM Extended High Performance Cartridge tape written in 256-track recording technology on an IBM TotalStorage Enterprise High Performance Tape System 3590 Model E subsystem.

  This value is returned from the drive and is used as an approximation that is close to the actual value.

Note: For tape volumes written using 18-track or 36-track format on IBM Cartridge System Tape or IBM Enhanced Capacity Cartridge System Tape on an IBM 3480, 3490, or 3490E Magnetic Tape Subsystem, the user can specify any capacity from 1 to 2 147 483 646 kilobytes of data. This is essentially a user-defined capacity, defined to OAM with SETOAM TAPECAPACITY statement of SYS1.PARMLIB(CBROAMxx). This statement enables the user to set capacities higher or lower than the standard capacities described above. If the user has specified a capacity with the SETOAM TAPECAPACITY statement that is higher than the tape volume is physically capable of handling, then the data is written to the tape volume until the natural end of volume is reached. In this case, the capacity value displayed is the value indicated on the SETOAM TAPECAPACITY statement, even though it is not possible to actually write to that capacity.

FRESPACE
: The available free space left for writing data in kilobyte units (1 kilobyte = 1 024 bytes) on the volume. This value reflects the reduction of the percent-full (PFULL) value associated with the storage group for this tape volume.

LSTBLKID
: The block ID of the last block written on the volume.

PFULL
: An indication of what percent of the tape has been written.

NUMLBLKS
: The number of logical blocks of data that OAM has written to the volume.

NUMLKBW
: The number of logical kilobytes of data that OAM has written to the volume. This includes OAM control information recorded on the tape volume as well as user object data.

NUMPKBW
: The number of physical kilobytes of data that has been physically recorded on the tape medium. This includes OAM control information recorded on the tape volume as well as user object data.

Note: If the data on a tape represented by a row in this table is being written in a compacted format, the number of logical KB of data on the tape and the number of physical KB of data on the tape may not be the same.

FULL An indication of whether the volume is considered full by OAM. The possible values of this column are listed below:

- **Y** The volume is considered full by OAM.
- **N** The volume is not full.

READABLE An indication of whether the volume is considered readable by OAM. The possible values are listed below:

- **Y** The volume is readable.
- **N** The volume is not readable.

WRITABLE An indication of whether the volume is considered writable by OAM.

- **Y** The volume is writable.
- **N** The volume is not writable. The WRITABLE column is set to N when a permanent data check occurs while writing data to the volume. This prevents the volume from being selected by OAM for the writing of additional objects.

INUSE An indication of whether the volume is in use by an OAM process. The possible values are listed below:

- **Y** The volume is in use by an OAM process.
- **N** The volume is not in use by an OAM process.

COPIED Reserved for future use.

AVOLSER Reserved for future use.

NUMLKBDE The number of logical kilobytes (KB) of data which have been deleted from the tape volume.

Note: This number is an **approximation**. Due to the fact that the application can issue a DB2 ROLLBACK for the OSREQ DELETE, and OSMC can issue a DB2 ROLLBACK after the TAPEVOL row update for NUMLKBDE has been submitted to the OAM address space, this number might be greater than the actual amount of data which has been deleted from this tape volume.

COMPACT The tape compaction indicator for this tape volume. The only valid values for this field are:

- **Y** The tape volume was written in compacted format.
- **N** The tape volume was written in noncompacted format.

MEMBER Indicates the name of the OAM within a parallel sysplex that is currently managing and controlling the tape volume.

- For tape volumes that are currently mounted and allocated on a tape drive for use by OAM, this member name is the member name of the OAM on the system to which the tape drive is online and allocated.
- For tape volumes that are not currently mounted and allocated on a tape drive for use by OAM, this member name is blank.

EPI This column contains the ERDS Physical Identifier (EPI) which indicates the real underlying device type that is used to write OAM objects to this volume. This column is used to assist in problem diagnosis in a mixed device environment where native and emulated devices coexist.

BKTYPE This indicates whether this volume is used for first or second backup copies of an object when the volume has a type of “B”, which indicates that it is a backup volume belonging to an Object Backup storage group.

Appendix D. OAM System Management Facility (SMF) Records

The following information provides details concerning the OAM System Management Facility records, which measure OAM performance at the OSREQ macro interface level. The SMF Type for all Subtypes listed is SMF Type 85. For an overview of this function, see “Measuring OAM Transaction Performance Using SMF” on page 180.

OAM SMF Record Header

The OAM SMF record header, as shown in Table 59, is at the beginning of each SMF record written by OAM:

Table 59. Header Format for OAM SMF Records

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	SMF85LEN	2	binary	Record length.
2 2	SMF85SEG	2	binary	Segment descriptor.
4 4	SMF85FLG	1	binary	System indicator. **BIT** **MEANING WHEN SET** **0** Reserved. **1** Subtypes are valid. **2** Reserved. **3** MVS/SP™ Version 4 and above. bits 3, 4, 5, and 6 are on. See note. **4** MVS/SP Version 3. bits 4, 5, and 6 are on. **5** MVS/SP Version 2. bits 5 and 6 are on. **6** VS2. bit 6 is on. **7** Reserved. **Note:** It is recommended that you use record type 30 to obtain the MVS product level.
5 5	SMF85RTY	1	binary	Record type (decimal 85, hexadecimal X'55').
6 6	SMF85TME	4	binary	Time since midnight in hundredths of a second that the record was presented to SMF.
10 A	SMF85DTE	4	binary	Date the record was presented to SMF in the form of 0cyydddF, where F is the sign.
14 E	SMF85SID	4	EBCDIC	System Identification (from SID parameter).
18 12	SMF85SSI	4	EBCDIC	Subsystem identification, contains ’OAM‘ for all OAM SMF records.
22 16	SMF85STY	2	binary	Record subtype.
24 18	SMF85TRN	2	binary	Number of triplets in this record. A triplet is a set of offset/length/number values that defines a section of the record.
26 1A	SMF85PSO	4	binary	Offset to OAM product section.
30 1E	SMF85PSL	2	binary	Length of OAM product section.
32 20	SMF85PSN	2	binary	Number of OAM product section.
34 22	SMF85OSO	4	binary	Offset to OAM subtype data section.
38 26	SMF85OSL	2	binary	Length of OAM subtype data section.
40 28	SMF85OSN	2	binary	Number of OAM subtype data sections.

Table 59. Header Format for OAM SMF Records (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
42 2A	*	6	binary	Reserved.

OAM SMF Record Product Section

Each OAM SMF record has a 112-byte OAM product section following the standard SMF record header. The OAM product section contains product identification information and common information to all OAM SMF record subtypes. If a field is not used for a particular subtype and the format of the field is shown as EBCDIC in the FORMAT column of the table describing the SMF record subtype, then the field contains EBCDIC blanks. If a field is not used for a particular subtype and the format of the field is shown as binary in the FORMAT column of the table describing the SMF record subtype, then the field contains binary zeros. Table 60 describes the format of the product section:

Table 60. Product Section Format for OAM SMF Subtypes

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	R85PCID	9	EBCDIC	Component ID for OAM. For DFSMS, this field contains the characters ’5695DF180’.
9 9	R85PVID	1	binary	Version number for DFSMS.
10 A	R85PRID	1	binary	Release number for DFSMS.
11 B	R85PMID	1	binary	Modification level for DFSMS.
12 C	R85PFMID	8	EBCDIC	SMP/E FMID for OAM.
20 14	*	4	binary	Reserved.
24 18	R85PCPUI	8	binary	CPU ID as stored by S/390 Store CPU ID (STIDP) instruction.
56 38	R85PPGMN	8	EBCDIC	Contains the job step program name. The job step program name is the name of the program specified on the job control language (JCL) EXEC statement with the PGM= keyword.
64 40	R85USRID	8	EBCDIC	User identification or blanks.
72 48	R85PTRXN	8	EBCDIC	Contains the transaction name for subtypes 2-6. The transaction names are specified as follows. For all other subtypes, this field contains blanks. **Environment** **Meaning** **CICS** The name of the CICS transaction that invoked the OSREQ macro. **IMS** The name of the IMS transaction that invoked the OSREQ macro. **OTHER** This field contains blanks, if the OSREQ macro was invoked from any other environment.
80 50	R85PSTRT	8	binary	Starting time of the function in 8-byte STCK format.
88 58	R85PENDT	8	binary	Ending time of the function in 8-byte STCK format.
96 60	R85PRESP	4	binary	Elapsed time of the function in milliseconds (.001 second units).
100 64	*	12	binary	Reserved.
Note: STCK = S/390 STORE CLOCK				

OSREQ Activity Subtypes 1–7 Data Section Format

The format of the subtype data section for all OSREQ macro functions is identical; although, not all of the fields are applicable to all OSREQ functions. Also, with the exception of the OSREQ return code (ST1RC), the rest of the fields are not valid if the OSREQ function fails. The following are subtypes and descriptions for the functions of the OSREQ macro:

1 OSREQ ACCESS
2 OSREQ STORE
3 OSREQ RETRIEVE
4 OSREQ QUERY
5 OSREQ CHANGE
6 OSREQ DELETE
7 OSREQ UNACCESS

Table 61 shows the format of the subtype data section for all OSREQ functions (subtypes 1–7):

Table 61. Subtype Data Section Format for OSREQ functions

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST1COLN	44	EBCDIC	Collection name. Valid for subtypes 2, 3, 4, 5, and 6.
44 2C	ST1OBJN	44	EBCDIC	Object name. Valid for subtypes 2, 3, 4, 5, and 6.
88 58	ST1SGN	8	EBCDIC	Storage group name. Valid for subtypes 2, 3, 4, 5, and 6.
96 60	ST1SCN	8	EBCDIC	Storage class name. Valid for subtypes 2, 4, and 5.
104 68	ST1MCN	8	EBCDIC	Management class name. Valid for subtypes 2, 4, and 5.
112 70	ST1OFF	4	binary	Offset for partial object retrieve (subtype 3). Zero for all others.
116 74	ST1LEN	4	binary	Length. Valid for subtypes 2, 3, 4, and 6. **SUBTYPE** **MEANING** **1** Unused, contains binary zero. **2** Length of object stored. **3** Number of bytes retrieved. **4** Number of QEL elements returned to the application program. **5** Unused, contains binary zero. **6** Length of object deleted. **7** Unused, contains binary zero.
120 78	ST1TTOK	16	binary	OSREQ tracking token supplied with TTOKEN keyword on the OSREQ macro. **Note:** Any application programs that want to use the new TTOKEN keyword interface need to be recompiled with the new OSREQ macro. For more information concerning the TTOKEN keyword, see Table 37 on page 211, and refer to *z/OS DFSMS OAM Application Programmer's Reference*.
136 88	ST1TOK	8	binary	OSREQ token.

Table 61. Subtype Data Section Format for OSREQ functions (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
144 90	ST1VSN	6	EBCDIC	Volume serial number. Valid for subtypes 2, 3, and 6. For an OSREQ STORE request (subtype 2), this field contains the volume serial number of the tape or optical volume to which the primary copy of the object was stored. Only valid if bit 1 or 2 is on in field ST2FLGS. For an OSREQ RETRIEVE request (subtype 3), this field contains the volume serial number of the tape or optical volume from which the copy of the object was retrieved. Either the first or the second backup copy is retrieved as determined by the VIEW=BACKUP | BACKUP2 option indicated on the RETRIEVE request. Valid if bit 1, 2, 3, 4, 5, or 6 is on in field ST3FLGS. For an OSREQ DELETE request (subtype 6), this field contains the volume serial number of the tape or optical volume from which the primary copy of the object was deleted. Valid if bit 1 or 2 is on in field ST6FLGS.

Table 61. Subtype Data Section Format for OSREQ functions (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
150 96	ST1VMT	2	EBCDIC	Volume media type. Valid for subtype 2, 3, and 6. If a volume serial number is contained in the previous field (ST1VSN), this field contains the media type of the volume whose volume serial number is in field ST1VSN as follows: **VALUE** **MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
152 98	ST1RC	4	binary	OSREQ return code. Value in register 15 following the OSREQ macro invocation.
156 9C	ST1RS	4	binary	OSREQ reason code. Value in register 0 following the OSREQ macro invocation.
160 A0	ST1FLGS	4	binary	Processing flags. For subtype 1, all bits contain zero.

Table 61. Subtype Data Section Format for OSREQ functions (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
160 A0	ST2FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 2, the following bit definitions apply: **BIT** **MEANING** **0** When on, the object is stored to DASD. **1** When on, the object is stored to optical. **2** When on, the object is stored to tape. **3** Unused. **4** Unused. **5** When on, the OSREQ STORE request resulted in the mounting of a shelf-resident removable media volume (tape or optical) by an operator. This bit is only valid if bit 1 or 2 is on. **6** When on, the OSREQ STORE request resulted in the mounting of a library-resident removable media volume (tape or optical) inside an automated storage library. This bit is only valid if bit 1 or 2 is on. **7** When on, the OSREQ STORE request was satisfied using an already mounted removable media volume (tape or optical). This bit is only valid if bit 1 or 2 is on.

Table 61. Subtype Data Section Format for OSREQ functions (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
160 A0	ST3FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below all bits are zero and reserved. For subtype 3, the following bit definitions apply: **BIT** **MEANING** **0** When on, the primary copy of the object was retrieved from DASD. **1** When on, the primary copy of the object was retrieved from optical. **2** When on, the primary copy of the object was retrieved from tape. **3** When on, either the first or the second backup copy of the object was retrieved from optical as a result of VIEW=BACKUP or VIEW=BACKUP2 being specified on the OSREQ macro. Refer to bit 10 to indicate which backup copy was retrieved. **4** When on, either the first or the second backup copy of the object was retrieved from tape as a result of VIEW=BACKUP or VIEW=BACKUP2 being specified on the OSREQ macro. Refer to bit 10 to indicate which backup copy was retrieved. **5** When on, either the first or the second backup copy of the object was retrieved from optical as a result of the primary copy of the object residing on an unreadable optical disk volume and the automatic access to backup being active. Refer to bit 10 to indicate which backup copy was retrieved. **6** When on, either the first or the second backup copy of the object was retrieved from tape as a result of the primary copy of the object residing on an unreadable optical disk volume and the automatic access to backup being active. Refer to bit 10 to indicate which backup copy was retrieved. **7** When on, the OSREQ RETRIEVE request resulted in the mounting of a shelf-resident removable media volume (tape or optical) by an operator. This bit is only valid if bit 1, 2, 3, 5, or 6 is on. **8** When on, the OSREQ RETRIEVE request resulted in the mounting of a library-resident removable media volume (tape or optical) inside an automated storage library. This bit is only valid if bit 1, 2, 3, 5, or 6 is on. **9** When on, the OSREQ RETRIEVE request was satisfied using an already mounted removable media volume (tape or optical). This bit is only valid of bit 1, 2, 3, 5, or 6 is on. **10** When on, the second backup copy of the object was retrieved.
160 A0	ST4FLGS	4	binary	Processing flags. For subtype 4, all bits contain zero.

Table 61. Subtype Data Section Format for OSREQ functions (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
160 A0	ST5FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 5, the following bit definitions apply: **BIT** **MEANING** **0** When on, management class is specified on the OSREQ CHANGE macro. **1** When on, storage class is specified on the OSREQ CHANGE macro. **2** When on, retention period is specified on the OSREQ CHANGE macro.
160 A0	ST6FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. Unless specified below, all bits are zero and reserved. For subtype 6, the following bit definitions apply: **BIT** **MEANING** **0** When on, the primary copy of the object is deleted from DASD. **1** When on, the primary copy of the object is deleted from optical. **2** When on, the primary copy of the object is deleted from tape. **3** When on, the first backup copy of the object is deleted from optical. **4** When on, the first backup copy of the object is deleted from tape. **5** When on, the second backup copy of the object is deleted from optical. **6** When on, the second backup copy of the object is deleted from tape.
160 A0	ST7FLGS	4	binary	Processing flags. For subtype 7, all bits contain zero.

Although subtypes 1–7 share a common subtype data section, not all fields are valid for each of the seven subtypes. Table 62 identifies which fields in the OAM subtype data section are valid for each of the seven OSREQ subtypes:

Table 62. Valid Subtype Data Section Fields for OSREQ Functions

FIELD NAME	OSREQ ACCESS Subtype 1	OSREQ STORE Subtype 2	OSREQ RETRIEVE Subtype 3	OSREQ QUERY Subtype 4	OSREQ CHANGE Subtype 5	OSREQ DELETE Subtype 6	OSREQ UNACCESS Subtype 7
ST1COLN		X	X	X	X	X	
ST1OBJN		X	X	X	X	X	
ST1SGN		X	X	X	X	X	
ST1SCN		X		X	X		
ST1MCN		X		X	X		
ST1OFF			X				
ST1LEN		X	X	X		X	
ST1TTOK	X	X	X	X	X	X	X
ST1TOK	X	X	X	X	X	X	X
ST1VSN		X	X			X	

Table 62. Valid Subtype Data Section Fields for OSREQ Functions (continued)

FIELD NAME	OSREQ ACCESS Subtype 1	OSREQ STORE Subtype 2	OSREQ RETRIEVE Subtype 3	OSREQ QUERY Subtype 4	OSREQ CHANGE Subtype 5	OSREQ DELETE Subtype 6	OSREQ UNACCESS Subtype 7
ST1VMT		X	X			X	
ST1RC	X	X	X	X	X	X	X
ST1RS	X	X	X	X	X	X	X
ST1FLGS		X	X		X	X	

OSMC Storage Management Activity (Subtypes 32–35)

Table 63 describes the format of the subtype data section for the following OAM SMF record subtypes:

32 OSMC Storage Group Processing
33 OSMC DASD Space Management Processing
34 OAM Volume Recovery Utility
35 OSMC Move Volume Utility

Table 63. Format of the Subtype Data Section for Subtypes 32–35

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST32SGN	8	EBCDIC	OBJECT or OBJECT BACKUP storage group name.
8 8	ST32VSN0	6	EBCDIC	Volume serial number of a tape or optical volume. Valid for subtypes 34 and 35. This field contains blanks for all other subtypes.
14 E	ST32VSN1	6	EBCDIC	Volume serial number of the opposite side of the optical volume. Valid for subtypes 34 and 35. If the volume serial number contained in field ST32VSN0 is the volume serial number of a tape volume this field contains blanks. This field contains blanks for all other subtypes.

Table 63. Format of the Subtype Data Section for Subtypes 32–35 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
20 14	ST32OMT	2	EBCDIC	Media type of the volume identified in field ST32VSN0. Valid for subtypes 34 and 35. This field contains blanks for all other subtypes. **VALUE MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
22 16	*	2	binary	Reserved.
24 18	ST32PDWO	4	binary	Number of primary objects written to DASD.
28 1C	ST32PDWK	4	binary	Number of kilobytes of primary object data written to DASD.
32 20	ST32PDRO	4	binary	Number of primary objects read from DASD.
36 24	ST32PDRK	4	binary	Number of kilobytes of primary object data read from DASD.
40 28	ST32PDDO	4	binary	Number of primary objects deleted from DASD.
44 2C	ST32PDDK	4	binary	Number of kilobytes of primary object data deleted from DASD.
48 30	ST32POWO	4	binary	Number of primary objects written to optical.
52 34	ST32POWK	4	binary	Number of kilobytes of primary object data written to optical.
56 38	ST32PORO	4	binary	Number of primary objects read from optical.
60 3C	ST32PORK	4	binary	Number of kilobytes of primary object data read from optical.
64 40	ST32PODO	4	binary	Number of primary objects deleted from optical.

Table 63. Format of the Subtype Data Section for Subtypes 32–35 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
68 44	ST32PODK	4	binary	Number of kilobytes of primary object data deleted from optical.
72 48	ST32PTWO	4	binary	Number of primary objects written to tape.
76 4C	ST32PTWK	4	binary	Number of kilobytes of primary object data written to tape.
80 50	ST32PTRO	4	binary	Number of primary objects read from tape.
84 54	ST32PTRK	4	binary	Number of kilobytes of primary object data read from tape.
88 58	ST32PTDO	4	binary	Number of primary objects deleted from tape.
92 5C	ST32PTDK	4	binary	Number of kilobytes of primary object data deleted from tape.
96 60	ST32BOWO	4	binary	Number of backup objects written to optical.
100 64	ST32BOWK	4	binary	Number of kilobytes of backup object data written to optical.
104 68	ST32BORO	4	binary	Number of backup objects read from optical.
108 6C	ST32BORK	4	binary	Number of kilobytes of backup object data read from optical.
112 70	ST32BODO	4	binary	Number of backup objects deleted from optical.
116 74	ST32BODK	4	binary	Number of kilobytes of backup object data deleted from optical.
120 78	ST32BTWO	4	binary	Number of backup objects written to tape.
124 7C	ST32BTWK	4	binary	Number of kilobytes of backup object data written to tape.
128 80	ST32BTRO	4	binary	Number of backup objects read from tape.
132 84	ST32PTRK	4	binary	Number of kilobytes of backup object data read from tape.
136 88	ST32BTDO	4	binary	Number of backup objects deleted from tape.
140 8C	ST32BTDK	4	binary	Number of kilobytes of backup object data deleted from tape.
144 90	ST32B2OWO	4	binary	Number of BACKUP2 objects written to optical.
148 94	ST32B2OWK	4	binary	Number of kilobytes of BACKUP2 objects written to optical.
152 98	ST32B2ORO	4	binary	Number of BACKUP2 objects read from optical.
156 9C	ST32B2ORK	4	binary	Number of rows kilobytes of BACKUP2 objects read from optical.
160 A0	ST32B2ODO	4	binary	Number of BACKUP2 objects deleted from optical.
164 A4	ST32B2ODK	4	binary	Number of kilobytes of BACKUP2 objects deleted from optical.
168 A8	ST32B2TWO	4	binary	Number of BACKUP2 objects written to tape.
172 AC	ST32B2TWK	4	binary	Number of kilobytes of BACKUP2 objects written to tape.
176 B0	ST32B2TRO	4	binary	Number of BACKUP2 objects read from tape.
180 B4	ST32B2TRK	4	binary	Number of kilobytes of BACKUP2 objects read from tape.

Table 63. Format of the Subtype Data Section for Subtypes 32–35 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
184 B8	ST32B2TDO	4	binary	Number of BACKUP2 objects logically deleted from tape.
188 BC	ST32B2TDK	4	binary	Number of kilobytes of BACKUP2 objects logically deleted from tape.
192 C0	ST32DTUP	4	binary	Number of rows updated in the object directory table.
196 C4	ST32DTDE	4	binary	Number of rows deleted from the object directory table.
200 C8	ST324KIN	4	binary	Number of rows inserted into the 4K object storage table.
204 CC	ST324KDE	4	binary	Number of rows deleted from the 4K object storage table.
208 D0	ST3232KI	4	binary	Number of rows inserted into the 32K object storage table.
212 D4	ST3232KD	4	binary	Number of rows deleted from the 32K object storage table.
216 D8	ST32NCE	4	binary	Number of optical cartridges expired. Valid only for Subtype 32.
220 DC	ST32FLGS	4	binary	Processing flags. **BIT** **MEANING** **0** When on, this process was invoked automatically under software control. **1** When on, this process was invoked by a MODIFY OAM,START command issued from an MVS console. **2** When on, this process was invoked using an ISMF line operator. **3** When on, volume recovery was invoked with the BACKUP1 keyword or defaulted to BACKUP1. **4** When on, volume recovery was invoked with the BACKUP2 keyword. **5–31** Reserved.

OSMC Single Object Recovery Utility (Subtype 36)

Table 64 describes the format of the subtype data section for a subtype 36 OAM SMF record for the single object recovery utility.

Table 64. Format of the Subtype Data Section for Subtype 36

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST36COLN	44	EBCDIC	Collection name.
44 2C	ST36CNID	4	binary	Collection ID.
48 30	ST36OBJN	44	EBCDIC	Object name.
92 5C	ST36SGN	8	EBCDIC	OBJECT storage group name.
100 64	ST37OLEN	4	binary	Object length.

Table 64. Format of the Subtype Data Section for Subtype 36 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
104 68	ST36BVSN	6	EBCDIC	Volume serial number of the optical or tape volume from which the backup copy of the object was read. The backup copy can be either the first or the second backup copy as determined by options specified on the F OAM,RECOVERY command. The options are: BACKUP1 | BACKUP2.
110 6E	ST36BMT	2	EBCDIC	Media type of volume from which the backup copy of the object was read: **VALUE MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
112 70	ST36BTKN	4	binary	Volume location token associated with the backup copy of the object on the volume specified in the ST36BVSN field.
116 78	ST36TVSN	6	EBCDIC	Volume serial number of the target optical or tape volume to which the new primary copy of the object was written. This field contains blanks if the new location is on DASD.

Table 64. Format of the Subtype Data Section for Subtype 36 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
122 7E	ST36TMT	2	EBCDIC	Media type of target optical or tape volume to which the new primary copy of the object was written. This field contains blanks if the new primary copy of the object was written to DASD: **VALUE MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
124 80	ST36OVSN	6	EBCDIC	Volume serial number of the original optical or tape volume on which the primary copy of the object resided prior to the start of the single object recovery utility. This field contains blanks if the original location was on DASD.

Table 64. Format of the Subtype Data Section for Subtype 36 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
130 82	ST36OMT	2	EBCDIC	Media type of the original optical or tape volume on which the primary copy of the object resided prior to the start of the single object recovery utility. This field contains blanks if the primary copy of the object currently resides on DASD: **VALUE** **MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **02** IBM 3480 Cartridge System Tape. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
132 84	ST36FLGS	4	binary	Processing flags. **BIT** **MEANING** **0** When on, object recovery was invoked with the BACKUP1 keyword or defaulted to BACKUP1. **1** When on, object recovery was invoked with the BACKUP2 keyword. **2–31** Reserved.

OSMC Library Space Management (Subtype 37)

Table 65 describes the format of the subtype data section for a subtype 37 OAM SMF record for OSMC library space management.

Table 65. Format of the Subtype Data Section for Subtype 37

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST37LIBN	8	EBCDIC	Library name.

Table 65. Format of the Subtype Data Section for Subtype 37 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
8 8	ST37LIBD	8	EBCDIC	Library device type.
16 10	ST37NOCE	4	binary	Number of optical disk cartridges ejected.
20 14	ST37FLGS	4	binary	Processing flags. **BIT** **MEANING** **0** When on, library space management is invoked automatically under software control due to a storage group out-of-space condition in the specified library. **1** When on, library space management is invoked by a **F OAM,START,LIBMGT** command issued from an MVS console. **2–31** Reserved.

LCS Optical Library/Drive Vary Online/Offline (Subtypes 64–67)

Table 66 describes the format of the subtype data section for the following subtypes:

64 LCS optical drive vary online.
65 LCS optical drive vary offline.
66 LCS optical library vary online.
67 LCS optical library vary offline.

Table 66. Format of the Subtype Data Section for Subtypes 64–67

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST64OLN	8	EBCDIC	Contains the real optical library name for the operator-accessible drive.
8 8	ST64OLDT	8	EBCDIC	Optical library device type.
16 10	ST64OLDN	4	EBCDIC	MVS device number corresponding to the optical library.
20 14	ST64ODN	8	EBCDIC	Optical drive name. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
28 1C	ST64ODDT	8	EBCDIC	Optical drive device type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
36 24	ST64ODDN	4	EBCDIC	MVS device number corresponding to the optical drive. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
40 28	ST64VSN0	6	EBCDIC	Volume serial number of the currently mounted volume. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
46 2E	ST64VSN1	6	EBCDIC	Volume serial number of the opposite side of the currently mounted volume. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.

Table 66. Format of the Subtype Data Section for Subtypes 64–67 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
52 34	ST64OMT	2	EBCDIC	Optical media type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **VALUE MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
54 36	ST64ODT	1	EBCDIC	Optical drive type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **VALUE MEANING** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.
55 37	ST64OVT	1	EBCDIC	Optical volume type. Valid for subtypes 64 and 65. This field contains blanks for other subtypes. **VALUE MEANING** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group. **S** Optical volume is a scratch volume.
56 38	ST64SGN	8	EBCDIC	Storage group name. Valid for subtypes 64 and 65. This field contains blanks for other subtypes.
64 40	ST64LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed.
68 44	ST64LDQT	4	binary	LCS dispatcher-queued time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed.
72 48	ST64LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds this request has spent on the LCS execution-queue being processed.

Table 66. Format of the Subtype Data Section for Subtypes 64–67 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
76 4C	ST64LTQT	4	binary	LCS library task queue time. The amount of time in milliseconds that this request has spent on the LCS library queue waiting to be processed. Normally, this field represents the cartridge transport mechanism wait time. That is, the time spent waiting for the cartridge transport mechanism within the automated optical disk library to become available. Valid for subtypes 66 and 67.
80 50	ST64LTPT	4	binary	LCS library task processing time. The amount of time in milliseconds that this request took to be processed by the library task. Normally, this field represents the cartridge transport mechanism service time. This is, the time spent by the cartridge transport mechanism within the automated optical disk library performing mechanical motion to move cartridges within the optical disk library. Valid for subtypes 66 and 67.
84 54	ST64RC	4	binary	LCS return code
88 58	ST64RS	4	binary	LCS reason code.
92 5C	ST64FLGS	4	binary	Processing flags. **BIT** **MEANING** **0–31** Reserved

Although subtypes 64–67 share a common subtype data section, not all fields are valid for each of the four subtypes. Table 67 identifies which fields in the OAM subtype data section are valid for each of the four subtypes.

Table 67. Valid Subtype Data Section Fields for Subtypes 64–67

FIELD NAME	DRIVE ONLINE Subtype 64	DRIVE OFFLINE Subtype 65	LIBRARY ONLINE Subtype 66	LIBRARY OFFLINE Subtype 67
ST64OLN	X	X	X	X
ST64OLDT	X	X	X	X
ST64OLDN	X	X	X	X
ST64ODN	X	X		
ST64ODDT	X	X		
ST64ODDN	X	X		
ST64VSN0	See table note 1.	See table note 2.		
ST64VSN1	See table note 1.	See table note 2.		
ST64OMT	See table note 1.	See table note 2.		
ST64ODT	X	X		
ST64OVT	See table note 1.	See table note 2.		
ST64SGN	See table note 1.	See table note 2.		
ST64LIQT				
ST64LDQT	X	X	X	X
ST64LTQT			X	X
ST64LTPT			X	X
ST64RC	X	X	X	X
ST64RS	X	X	X	X

Table 67. Valid Subtype Data Section Fields for Subtypes 64–67 (continued)

FIELD NAME	DRIVE ONLINE Subtype 64	DRIVE OFFLINE Subtype 65	LIBRARY ONLINE Subtype 66	LIBRARY OFFLINE Subtype 67
ST64FLGS	X	X	X	X

Note:
1. This field is only valid if there is an optical disk cartridge mounted in the drive at the time the **VARY SMS,DRIVE(***drive_name***),ONLINE** is issued.
2. This field is only valid if there is an optical disk cartridge mounted in the drive at the time the **VARY SMS,DRIVE(***drive_name***),OFFLINE** is issued.

LCS Optical Cartridge Entry, Eject, Label, Audit, Mount, and Demount (Subtypes 68–73)

Table 68 describes the format of the subtype data section for the following OAM SMF record subtypes:

68 LCS optical cartridge entry
69 LCS optical cartridge eject
70 LCS optical cartridge label
71 LCS optical volume audit
72 LCS optical volume mount
73 LCS optical volume demount

Table 68. Format of the Subtype Data Section for Subtypes 68–73

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST68OLN	8	EBCDIC	Optical library name. This field contains the real library name of an operator-accessible drive.
8 8	ST68OLDT	8	EBCDIC	Optical library device type.
16 10	ST68OLDN	4	EBCDIC	MVS device number that corresponds to the optical library.
20 14	ST68ODN	8	EBCDIC	Optical drive name.
28 1C	ST68ODDT	8	EBCDIC	Optical drive device type.
36 24	ST688ODDN	4	EBCDIC	MVS device number that corresponds to the optical drive.
40 28	ST68VSN0	6	EBCDIC	Volume serial number.
46 2E	ST68VSN1	6	EBCDIC	Volume serial number of the opposite side of the optical disk.

Table 68. Format of the Subtype Data Section for Subtypes 68–73 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
52 34	ST68OMT	2	EBCDIC	Optical media type **VALUE MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
54 36	ST68ODT	1	EBCDIC	Optical drive type: **VALUE MEANING** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.
55 37	ST68OVT	1	EBCDIC	Optical volume type: **VALUE MEANING** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group. **S** Optical volume is a scratch volume.
56 38	ST68SGN	8	EBCDIC	Storage group name.
64 40	ST68LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed.
68 44	ST68LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed.
72 48	ST68LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds this request has spent on the LCS execution-queue being processed.

Table 68. Format of the Subtype Data Section for Subtypes 68–73 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
76 4C	ST68LTQT	4	binary	LCS library task queue time. The amount of time in milliseconds this request has spent on the LCS library queue waiting to be processed. Normally, this field represents the cartridge transport mechanism wait time. That is, the time spent waiting for the cartridge transport mechanism within the automated optical disk library to become available.
80 50	ST68LTPT	4	binary	LCS library task processing time. The amount of time in milliseconds this request took to be processed by the library task. Normally, this field represents the cartridge transport mechanism service time. That is, the time spent by the cartridge transport mechanism within the automated optical disk library performing mechanical motion to move cartridges within the optical disk library.
84 54	ST68RC	4	binary	LCS return code.
88 58	ST68RS	4	binary	LCS reason code.
92 5C	ST68FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 68: **BIT** **MEANING** **0** When on, the volume serial number described by field ST68VSN0 required formatting as part of the optical cartridge entry processing. **1** When on, the volume serial number described by field ST68VSN1 required formatting as part of optical cartridge entry processing. **2** When on, the volume serial number described by field ST68VSN0 was not known to OAM at the time of being entered into the optical library. There was no row for this optical disk volume in the Volume table in the OCDB. **3** When on, the volume serial number described by field ST68VSN1 was not known to OAM at the time of being entered into the optical library. There was no row for this optical disk volume in the Volume table in the OCDB. **4–31** Reserved.
92 5C	ST69FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 69: **BIT** **MEANING** **0** When on, this optical cartridge was automatically ejected by the system due to an error condition known as a system-initiated eject request. **1–31** Reserved.
92 5C	ST70FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 70: **BIT** **MEANING** **0–31** Reserved.

Table 68. Format of the Subtype Data Section for Subtypes 68–73 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
92 5C	ST71FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 71: **BIT** **MEANING** **0–31** Reserved.
92 5C	ST72FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 72: **BIT** **MEANING** **0** When on, the volume serial number described by field ST68VSN0 required formatting as part of the optical volume mount processing. **1** When on, the volume serial number described by field ST68VSN1 required formatting as part of the optical volume mount processing. **2–31** Reserved.
92 5C	ST73FLGS	4	binary	Processing flags. The meaning is dependent on the record subtype. The following bit definitions apply for record subtype 73: **BIT** **MEANING** **0–31** Reserved.
96 60	ST68TMNT	4	binary	Elapsed time in milliseconds that the optical disk volume was mounted. Valid for subtypes 69 and 73.
100 64	ST68NOW	4	binary	Number of objects written to this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
104 68	ST68NKBW	4	binary	Number of kilobytes of object data written to this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
108 6C	ST68NOR	4	binary	Number of objects read from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
112 70	ST68NKBR	4	binary	Number of kilobytes of object data read from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
116 74	ST68NOD	4	binary	Number of objects deleted from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.
120 78	ST68NKBD	4	binary	Number of kilobytes of object data deleted from this optical disk volume while it was mounted. Valid for subtypes 69 and 73.

Although subtypes 68–73 share a common subtype data section, not all fields are valid for each of the six subtypes. Table 69 identifies which fields in the OAM subtype data section are valid for each of the six subtypes.

Table 69. Valid Subtype Data Section Fields for Subtypes 68–73

FIELD NAME	OPTICAL CARTRIDGE ENTRY Subtype 68	OPTICAL CARTRIDGE EJECT Subtype 69	OPTICAL CARTRIDGE LABEL Subtype 70	OPTICAL VOLUME AUDIT Subtype 71	OPTICAL VOLUME MOUNT Subtype 72	OPTICAL VOLUME DEMOUNT Subtype 73
ST68OLN	X	X	X	X	X	X
ST68OLDT	X	X	X	X	X	X

Table 69. Valid Subtype Data Section Fields for Subtypes 68–73 (continued)

FIELD NAME	OPTICAL CARTRIDGE ENTRY Subtype 68	OPTICAL CARTRIDGE EJECT Subtype 69	OPTICAL CARTRIDGE LABEL Subtype 70	OPTICAL VOLUME AUDIT Subtype 71	OPTICAL VOLUME MOUNT Subtype 72	OPTICAL VOLUME DEMOUNT Subtype 73
ST68OLDN	X	X	X	X	X	X
ST68ODN	X	See table note.	X	X	X	X
ST68ODDT	X	See table note.	X	X	X	X
ST68ODDN	X	See table note.	X	X	X	X
ST68VSN0	X	X	X	X	X	X
ST68VSN1	X	X	X	X	X	X
ST68OMT	X	X	X	X	X	X
ST68ODT	X	See table note.	X	X	X	X
ST68OVT	X	X	X	X	X	X
ST68SGN	X	X	X	X	X	X
ST68LIQT		X	X	X		
ST68LDQT	X	X	X	X		
ST668LEQT	X	X	X	X		
ST68LTQT		X		X		
ST68LTPT		X		X		
ST68RC	X	X	X	X	X	X
ST68RS	X	X	X	X	X	X
ST68FLGS	X	X	X	X	X	X
ST68TMNT		See table note.				X
ST68NOW		See table note.				X
ST68NKBW		See table note.				X
ST68NOR		See table note.				X
ST68NKBR		See table note.				X
ST68NOD		See table note.				X
ST68NKBD		See table note.				X
Note: This field contains valid data if the optical disk volume being ejected is mounted in an optical disk drive at the time that the **LIBRARY,EJECT,***volser* command or **F OAM,EJECT,***volser* command is received to eject the volume.						

LCS Optical Write, Read, Logical Delete, Physical Delete (Subtypes 74–77)

Table 70 on page 498 describes the format of the subtype data section for the following OAM SMF record subtypes:

74 LCS optical write request
75 LCS optical read request
76 LCS optical delete request (logical)
77 LCS optical delete request (physical)

Table 70. Format of the Subtype Data Section for Subtypes 74–77

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST74ORMN	16	EBCDIC	OAM request member name. Valid for subtypes 74 and 75.
16 10	ST74OTMN	16	EBCDIC	OAM target member name. Valid for subtypes 74 and 75.
32 20	ST74OLN	8	EBCDIC	Optical library name.
40 28	ST74OLDT	8	EBCDIC	Optical library device type.
48 30	ST74OLDN	4	EBCDIC	MVS device number that corresponds to the optical library.
52 34	ST74ODN	8	EBCDIC	Optical drive name.
60 3C	ST74ODDT	8	EBCDIC	Optical drive device type.
68 44	ST748ODDN	4	EBCDIC	MVS device number that corresponds to the optical drive.
72 48	ST74ODT	1	EBCDIC	Optical drive type: **VALUE** **MEANING** **L** Optical drive is a library-resident drive. **S** Optical drive is a stand-alone or operator-accessible drive.
73 49	ST74OVT	1	EBCDIC	Optical volume type **VALUE** **MEANING** **B** Optical volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Optical volume is a grouped volume belonging to an OBJECT storage group.
74 4A	ST74SGN	8	EBCDIC	Storage group name.
82 52	ST74LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds that this request has spent on the LCS input-work-queue waiting for processing.
86 56	ST74LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds that this request has spent on the LCS dispatcher-queue waiting for processing.
90 5A	ST74LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds that this request has spent on the LCS execution-queue processing.
94 5E	ST74LXQT	4	binary	XCF cross system processing-queue time. The amount of time in milliseconds that this request spent being processed on the XCF cross system queue. For subtypes 76 and 77, this field contains binary zeros.
98 62	ST74OVMT	4	binary	Optical volume mount time. The amount of time in milliseconds that it took to mount the optical disk volume required by this request. This field is valid if bit 1, 2, or 3 in field ST74FLGS is on. This field is valid for subtypes 74, 75, and 77.
102 66	ST74OVDT	4	binary	Optical volume demount time. This is the amount of time in milliseconds that it took to demount the optical disk volume that was mounted prior to mounting the optical disk volume required by this request. The field is valid if bit 3 in field ST74FLGS is on. This field is valid for subtypes 74, 75, and 77.

Table 70. Format of the Subtype Data Section for Subtypes 74–77 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
106 6A	ST74FLGS	4	binary	Processing flags **BIT** **MEANING** **0** This request was processed using a currently mounted optical disk volume and did not require an unmounted optical disk volume to be mounted. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **1** This request was processed using the opposite side of a currently mounted optical disk volume. Therefore, this request required the optical disk volume to be turned over in order to access the volume on the opposite side of the currently mounted volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **2** This request required an unmounted optical disk volume to be mounted and the optical disk drive that was used to process this request was empty at the time of the request. Therefore, this request did not require a currently mounted optical disk volume to be demounted prior to mounting the required optical disk volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **3** This request required an unmounted optical disk volume to be mounted and the optical disk drive that was selected for this request was full. Therefore, this request required a currently mounted optical disk volume to be demounted prior to mounting the required optical disk volume. Valid for subtypes 74, 75, and 77. Not valid for subtype 76. **4–31** Reserved.
110 6E	ST74NOBJ	4	binary	Total number of objects in this request. The maximum possible for this field is 280. With 280 object entries, the maximum SMF record size is 32 744 bytes.
114 72	ST74NKBP	4	binary	Total number of kilobytes of object data in this request.
118 76	ST74SOBJ	4	binary	Total number of objects in this request that processed successfully.
122 7A	ST74SKBP	4	binary	Total number of kilobytes of object data in this request that processed successfully.
126 7E	*	14	binary	Reserved.
Note: The following fields (comprising 116 bytes) are repeated for each object in the chained request, that is for the number of objects specified in the ST74NOBJ field. The maximum number of times that the following fields will be repeated is 280. With 280 object entries, the maximum SMF record size is 32 744 bytes.				
140 8C	ST74COLN	44	EBCDIC	Collection name.
184 B8	ST74OBJN	44	EBCDIC	Object name.
228 E4	ST74OLEN	4	binary	Object length.
232 E8	ST74OOFF	4	binary	Object offset. Valid for a subtype 75 partial object read.
236 EC	ST74VSN	6	EBCDIC	Volume serial number.

Table 70. Format of the Subtype Data Section for Subtypes 74–77 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
242 F2	ST74OMT	2	EBCDIC	Optical media type: **VALUE** **MEANING** **00** IBM 9247 12-inch 2000-MB optical disk media. **01** IBM 3995 5.25-inch 650-MB rewritable optical disk media. **03** IBM 3995 5.25-inch 650-MB WORM optical disk media. **11** IBM 3995 5.25-inch 1300-MB rewritable optical disk media. **13** IBM 3995 5.25-inch 1300-MB WORM optical disk media. **15** IBM 3995 5.25-inch 1300-MB CCW optical disk media. **21** IBM 3995 5.25-inch 2600-MB rewritable optical disk media. **23** IBM 3995 5.25-inch 2600-MB WORM optical disk media. **25** IBM 3995 5.25-inch 2600-MB CCW optical disk media. **31** IBM 3995 5.25-inch 5.2-GB rewritable optical disk media. **33** IBM 3995 5.25-inch 5.2-GB WORM optical disk media. **35** IBM 3995 5.25-inch 5.2-GB CCW optical disk media. **Note:** CCW = continuous composite WORM media. WORM = write-once-read-many.
244 F4	ST74OTKN	4	binary	Object volume location token.
248 F8	ST74RC	4	binary	LCS return code.
252 FC	ST74RS	4	binary	LCS reason code.

Although subtypes 74–77 share a common subtype data section, not all fields are valid for each of the four subtypes. Table 71 identifies which fields in the OAM subtype data section are valid for each of the four subtypes.

Table 71. Valid Subtype Data Section Fields for Subtypes 74–77

FIELD NAME	OPTICAL WRITE REQUEST Subtype 74	OPTICAL READ REQUEST Subtype 75	OPTICAL DELETE REQUEST (LOGICAL) Subtype 76	OPTICAL DELETE REQUEST (PHYSICAL) Subtype 77
ST74ORMN	X	X		
ST74OTMN	X	X		
ST74OLN	X	X	X	X
ST74OLDT	X	X	X	X
ST74OLDN	X	X	X	X
ST74ODN	X	X		X
ST74ODDT	X	X		X
ST74ODDN	X	X		X
ST74ODT	X	X		X
ST74OVT	X	X	X	X
ST74SGN	X	X	X	X

Table 71. Valid Subtype Data Section Fields for Subtypes 74–77 (continued)

FIELD NAME	OPTICAL WRITE REQUEST Subtype 74	OPTICAL READ REQUEST Subtype 75	OPTICAL DELETE REQUEST (LOGICAL) Subtype 76	OPTICAL DELETE REQUEST (PHYSICAL) Subtype 77
ST74LIQT	X	X	X	
ST74LDQT	X	X		
ST74LEQT	X	X		X
ST74LXQT	X	X		
ST74LVMT	X	X		X
ST74LVDT	X	X		X
ST74NOBJ	X	X	X	X
ST74NKBP	X	X	X	X
ST74FLGS	X	X	X	X
ST74SOBJ	X	X	X	X
ST74SKBP	X	X	X	X
ST74COLN	X	X	X	X
ST74OBJN	X	X	X	X
ST74OLEN	X	X	X	X
ST74OOFF		X		
ST74VSN	X	X	X	X
ST74OMT	X	X	X	X
ST74OTKN	X	X	X	X
ST74RC	X	X	X	X
ST74RS	X	X	X	X

LCS Tape Write and Read Request (Subtypes 78–79, and 88)

Table 72 describes the format of the subtype data section for the following OAM SMF record subtypes:

78 LCS tape write request
79 LCS tape read request
88 LCS object tape logical delete request

Table 72. Format of Subtype Data Section for Subtypes 78–79

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST78ORMN	16	EBCDIC	OAM request member name. Valid for subtype 79 only.
16 10	ST78OTMN	16	EBCDIC	OAM target member name. Valid for subtype 79 only.
32 20	ST78TDUN	8	EBCDIC	Tape drive unit name.
40 28	ST78TDDN	4	EBCDIC	MVS device number of the tape drive.
44 2C	ST78TVT	1	EBCDIC	OAM tape volume type. **VALUE MEANING** **B** Tape volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Tape volume is a grouped volume belonging to an OBJECT storage group.
45 2D	*	3	binary	Reserved.
48 30	ST78SGN	8	EBCDIC	Name of the OBJECT or OBJECT BACKUP storage group to which the tape volume belongs.
56 38	ST78LIQT	4	binary	LCS input-work-queue time. The amount of time in milliseconds this request has spent on the LCS input-work-queue waiting to be processed. For subtype 88, this field contains binary zeros.
60 3C	ST78LDQT	4	binary	LCS dispatcher-queue time. The amount of time in milliseconds this request has spent on the LCS dispatcher-queue waiting to be processed. For subtype 88, this field contains binary zeros.
64 40	ST78LEQT	4	binary	LCS execution-queue time. The amount of time in milliseconds that this request has spent on the LCS execution-queue being processed. For subtype 88, this field contains binary zeros.
68 44	ST78LXQT	4	binary	XCF cross system processing-queue time. The amount of time in milliseconds that this request has spent being processed on the XCF cross system queue. For subtypes 78 and 88, this field contains binary zeros.
72 48	ST78LMAT	4	binary	MVS dynamic allocation time. This is the amount of time in milliseconds that was required by MVS dynamic allocation (SVC 99) to dynamically allocate the tape drive. For subtypes 78 and 79, this field is only valid if bit 1 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
76 4C	ST78LMDT	4	binary	MVS dynamic deallocation time. This is the amount of time in milliseconds that was required by MVS dynamic deallocation (SVC99) to dynamically deallocate the tape drive. For subtypes 78 and 79, this field is only valid if bit 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.

Table 72. Format of Subtype Data Section for Subtypes 78–79 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
80 50	ST78LDCT	4	binary	DFP CLOSE time. This is the amount of time in milliseconds that was required by DFP CLOSE processing to close an already-opened tape data set. For subtypes 78 and 79, this field is only valid if bit 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
84 54	ST78LDOT	4	binary	DFP OPEN time. This is the amount of time in milliseconds that was required by DFP OPEN processing to open the tape data set. For subtypes 78 and 79, this field is only valid if bit 1 or 2 in field ST78FLGS is on. For subtype 88, this field contains binary zeros.
88 58	ST78LDPT	4	binary	DFP POINT time. This is the amount of time in milliseconds that was required by DFP POINT processing to position to the correct block-ID on the tape media. For subtype 88, this field contains binary zeros.
92 5C	ST78LBRT	4	binary	BSAM READ time. This is the amount of time in milliseconds that OAM spent in BSAM READ processing reading data from the tape volume. Valid for subtype 79.
96 60	ST78LBWT	4	binary	BSAM WRITE time. This is the amount of time in milliseconds that OAM spent in BSAM WRITE processing writing data to the tape volume. Valid for subtype 78.
100 64	ST78LBCT	4	binary	BSAM CHECK time. This is the amount of time in milliseconds that OAM spent in BSAM CHECK processing waiting for I/O operations to the tape volume to complete. Valid for subtype 78 and 79.
104 68	ST78FLGS	4	binary	Processing flags. For subtype 88, this field contains binary zeros. **BIT** **MEANING** **0** This request was processed using a currently mounted tape volume and did not require an unmounted tape volume to be mounted. **1** This request required an unmounted tape volume to be mounted and the tape drive that was used to process this request was empty at the start of processing this request. Therefore, this request did not require a currently mounted tape volume to be demounted prior to mounting the required tape volume. **2** This request required an unmounted tape volume to be mounted and the tape drive that was used to process this request was full at the start of processing this request. Therefore, this request required a currently mounted tape volume to be demounted prior to mounting the required tape volume. **3** This request was processed using a tape drive inside an automated tape library dataserver. **4–31** Reserved

Table 72. Format of Subtype Data Section for Subtypes 78–79 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
108 6C	ST78NOBJ	4	binary	Total number of objects in this request. The maximum value for this field is 280. With 280 object entries in this record, the maximum SMF record size for subtype 78 is 32 744 bytes.
112 70	ST78NKBP	4	binary	Total number of kilobytes of object data in this request.
116 74	ST78SOBJ	4	binary	Total number of objects in this request that processed successfully.
120 78	ST78SKBP	4	binary	Total number of kilobytes of object data in this request that processed successfully.
124 7C	*	16	binary	Reserved.
Note: The following fields (comprising 116 bytes) are repeated for each object in the chained request, that is for the number of objects specified in the ST78NOBJ field. The maximum number of times that the following fields will be repeated is 280. With 280 object entries, the maximum SMF record size for subtype 78 is 32 744 bytes.				
140 8C	ST78COLN	44	EBCDIC	Collection name.
184 B8	ST78OBJN	44	EBCDIC	Object name.
228 E4	ST78OLEN	4	binary	Object length.
232 E8	ST78OOFF	4	binary	Object offset. Valid for a subtype 79 partial object read.
236 EC	ST78VSN	6	EBCDIC	Volume serial number.
242 F2	ST78TMT	2	EBCDIC	Tape media type: **VALUE MEANING** **02** IBM 3480 Cartridge System Tape. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape.
244 F4	ST78OTKN	4	binary	Object volume location token.
248 F8	ST78RC	4	binary	LCS return code.
252 FC	ST78RS	4	binary	LCS reason code.

OAM Tape Volume Demount (Subtype 87)

Table 73 describes the format of the subtype 87 data section.

Table 73. Format of Subtype Data Section for Subtype 87

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
0 0	ST87TDDN	4	EBCDIC	MVS device number that corresponds to the tape drive on which the volume was mounted.
4 4	ST87TDDT	4	EBCDIC	MVS UCB device type associated with the tape drive on which the volume was mounted.
8 8	ST87TVUN	8	EBCDIC	Unit name associated with the tape volume and used to allocate the tape drive.
16 10	ST87VSN	6	EBCDIC	Volume serial number of the tape volume.

Table 73. Format of Subtype Data Section for Subtype 87 (continued)

OFFSETS	NAME	LENGTH	FORMAT	DESCRIPTION
22 16	ST87TMT	2	EBCDIC	Tape media type. **VALUE** **MEANING** **02** IBM 3480 Cartridge System Tape. **04** IBM 3480 Enhanced Capacity Cartridge System Tape. **05** IBM High Performance Cartridge Tape. **06** IBM Extended High Performance Cartridge Tape.
24 18	ST87TVT	1	EBCDIC	OAM tape volume type. **VALUE** **MEANING** **B** Tape volume is a backup volume belonging to an OBJECT BACKUP storage group. **G** Tape volume is a grouped volume belonging to an OBJECT storage group.
25 19	*	3	binary	Reserved.
28 1C	ST87SGN	8	EBCDIC	Name of the OBJECT or OBJECT BACKUP storage group.
36 24	ST87RC	4	binary	LCS return code.
40 28	ST87RS	4	binary	LCS reason code.
44 2C	ST87FLGS	4	binary	Processing flags. **BIT** **MEANING** **0–31** Reserved.
48 30	ST87TMNT	4	binary	Elapsed time in milliseconds that the tape volume was mounted, measured from the time that the first DFP OPEN macro completed to the time the tape volume was unallocated via a SVC 99 dynamic unallocation request.
52 34	ST87NOW	4	binary	Number of objects written to this tape volume while it was mounted.
56 38	ST87NKBW	4	binary	Number of logical kilobytes of object data written to this tape volume while it was mounted.
60 3C	ST87NOR	4	binary	Number of objects read from this tape volume while it was mounted.
64 40	ST87NKBR	4	binary	Number of kilobytes of object data read from this tape volume while it was mounted.

Invoking the SMF PARMLIB Member

The MVS operator can dynamically change which SMF records and record subtypes are currently being recorded by one of two methods:

- Issue a **SET SMF** = *xx* command at an MVS console to activate a new SMF PARMLIB member. The xx identifies the **SMFPRM***xx* member of the SYS1.PARMLIB that is to be activated by SMF.
- Issue a **SETSMF** command at an MVS system console to add a SUBPARM parameter or to replace any previously specified parameter in the active SMF PARMLIB member except for the ACTIVE, PROMPT, SID, or EXITS parameters.

For more information regarding the SET SMF, SETSMF, and the SMF PARMLIB member, refer to *z/OS MVS System Management Facilities (SMF)*; *z/OS MVS System Commands*; and *z/OS MVS Initialization and Tuning Reference*.

Changing SMF Recording

The MVS system operator or MVS system programmer can dynamically change OAM SMF recording using one of the following two methods:

- Update the SMF PARMLIB member (SMFPRMxx) to include the OAM SMF record subtypes:

  SYS(TYPE(85(2:3)))

  and activate the SMF PARMLIB member (SMFPRMxx) by entering the following MVS operator SET command:

  SET SMF=*xx*

  Note: The above example activates the OAM SMF recording for subtypes 2 and 3.
- Update the SMF options dynamically by entering the following MVS operator SETSMF command:

  SETSMF SYS(TYPE(85(4:6)))

  Note: The above example activates the OAM SMF recording for subtypes 4, 5, and 6.

For more information regarding the SET SMF and SETSMF commands refer to *z/OS MVS System Commands*.

Below are several examples of the format of the SETSMF command to activate various OAM SMF record subtypes from an MVS console:

- To exclude collecting all OAM SMF records, enter the following command:

  SETSMF SYS(NOTYPE(85)))
- To activate all OAM SMF record subtypes for the OSREQ macro application programming interface (subtypes 1–7), enter the following command:

  SETSMF SYS(TYPE(85(1:7)))
- To activate the OAM SMF record subtype for the OSMC storage group processing (subtype 32), enter the following command:

  SETSMF SYS(TYPE(85(32)))
- To activate the OAM SMF record subtype for the OSREQ RETRIEVE (subtype 3), LCS optical volume mount (subtype 72), and LCS optical read request (subtype 75), enter the following command:

  SETSMF SYS(TYPE(85(3,72,75)))

  To activate the OAM SMF record subtypes to track all optical library subsystem activity: optical cartridge entry (subtype 68), optical cartridge eject (subtype 69), optical volume audit (subtype 71), optical volume mount (subtype 72) and optical volume demount (subtype 73), enter the following command:

  SETSMF SYS(TYPE(85(68,69,71,72,73)))

- To activate the OAM SMF record subtypes to track all objects being retrieved from tape: OSREQ RETRIEVE (subtype 3) and LCS tape read requests (subtype 79), enter the following command:

  ```
  SETSMF  SYS(TYPE(85(3,79)))
  ```

DASD Space Allocation

The number of OAM SMF records written to the SMF data sets is dependent on two major factors:
- The amount of OAM activity that occurs on the processor complex, and
- The OAM SMF record subtypes that the system programmer has selected to be recorded in the SMF data sets.

Depending on the number of SMF records being recorded in the SMF data sets, the system programmer should perform the following activities:
- Determine which OAM SMF record subtypes should be captured.
- Make a preliminary determination of the number of each OAM SMF record subtype that will occur each hour or day.
- Calculate the additional DASD space requirements needed for SMF data sets based on the above two factors. The OAM SMF record subtype sizes are provided in the Table 29 on page 181. Variables and formulas for determining DASD requirements can be found in Table 7 on page 59, Table 8 on page 59, and Table 9 on page 60.
- Determine if the DASD space allocation quantities for the existing SMF data sets will be satisfactory given the additional space required by the OAM SMF records. If the current DASD space allocation quantity for the existing SMF data sets is insufficient based on how frequently the system programmer wishes SMF to switch data sets, then one or both of the following should be performed:
  – Reallocate the SMF data set with a larger primary space allocation quantity
  – Increase the number of SMF data sets
- Determine the adequacy of the DASD subsystem containing the SMF data sets to determine if the additional I/O activity caused by OAM recording the selected SMF records is going to introduce unacceptable levels of utilization and I/O contention on the DASD subsystem components, including:
  – DASD device level I/O contention
  – Control unit I/O contention
  – Channel path contention and utilization

Appendix E. Auto-Delete Installation Exit

This appendix contains product-sensitive programming interface and associated guidance information that describes how to tailor the OAM auto-delete installation exit to suit your needs.

Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* for information on the following tape library-related exits:
- Cartridge Entry Installation Exit (CBRUXENT)
- Change Use Attribute Installation Exit (CBRUXCUA)
- Cartridge Eject Installation Exit (CBRUXEJC)
- Volume Not in Library Installation Exit (CBRUXVNL)

Auto-Delete Installation Exit (CBRHADUX)

You can use the auto-delete installation exit (CBRHADUX) to confirm or bypass automatic deletion of objects during an OSMC management cycle in your OAM system.

The installation exit executes as part of the OAM Storage Management Component (OSMC) mainline processing. OSMC can delete an object when its lifetime expires in accordance with the definition of the management class assigned to the object. An object can also expire through an explicit expiration date. If the object has an explicit expiration date, that takes precedence over the defined management class for the object. Before any object is deleted, OSMC calls the auto-delete installation exit to approve or deny the request for object deletion. This approval or disapproval for object deletion is dependent upon the return code returned by the installation exit. The installation exit also records the deletion of the object so that other applications are kept in synchronization with the OAM directory.

Note: The installation exit is not called when a user requests deletion of an object during an OSREQ macro call. Refer to *z/OS DFSMS OAM Application Programmer's Reference* for information on how to use the OSREQ macro in your application.

OSMC performs the storage management cycle using a separate task for each storage group. This multitasking aspect means there may be concurrent execution of the auto-delete installation exit; therefore, it must be reentrant.

The installation exit is called for every object that is to be deleted by OSMC and one last time when there are no more objects to delete. This last call does not include object deletion information.

Installing and Replacing the CBRHADUX Installation Exit

You can use and modify the sample auto-delete installation exit that is provided in SAMPLIB (see “Sample Auto-Delete Installation Exit” on page 512), or you can write a new installation exit. If you modify or write your own auto-delete installation exit, you must reassemble the data set, and link-edit the control section (CSECT) into SYS1.LINKLIB. The name of the load module must be CBRHADUX.

In an OAMplex, when a new CBRHADUX load module is installed, make sure that changes are consistent on all OAMs within the OAMplex.

Writing the CBRHADUX Exit

In general, the routines you code for the auto-delete installation exit should follow these criteria:

- Written in Assembler H or High Level Assembler
- Handle multiple requests (reentrant)
- Reside in SYS1.LINKLIB
- Any valid combination of AMODE and RMODE
- Return to the caller using the BSM instruction

The sample auto-delete installation exit has a "Sample Auto-Delete Installation Exit" on page 512 that prevents objects from being deleted. It also contains code to allow automatic deletion of objects which can be activated with a simple code modification. Once this simple code modification has been made, the sample auto-delete installation exit reads a data set that contains a list of all the objects that should not be deleted. This data set is created by the installation when it finds certain objects that should be retained for longer periods of time. The data set should be small because finding these objects that need to be kept for a longer retention period happens infrequently. The data set records are read and the object names are stored in an in-storage table known as the verify table, which is referenced for each call. If the object appears in the verify table, the "do not delete" return code is sent back to the caller.

Note: Take care to avoid processing overhead because it can affect the time it takes to process the OSMC storage management cycle. For example, when the two lines of bypass code are removed, the sample exit, "Sample Auto-Delete Installation Exit" on page 512, reads the names of the objects once per cycle. It maintains the verify table to avoid I/O for each object, and it issues I/O to a data set instead of a teleprocessing link to allow for notification of deleted objects.

If the object name is *not* found in the verification table, the exit approves deletion. When the object is deleted, the name is written to a sequential data set called the notify data set. Fully qualified data set names contain both the object name and the collection name, allowing uniqueness across OSMC and concurrent I/O from the different tasks. A concatenation of these data sets (one for each group) provides input to other applications that need to synchronize their directories with the OAM object directories.

The sample uses the auto-delete installation field (ADUUFLD) to store a pointer to a dynamic area. Dynamic areas contain: save areas, system services parameter lists, and data control blocks (DCBs) that require updating.

The sample executes in 24-bit addressing mode. It can execute in 31-bit mode if you modify it to remove 24-bit dependencies. Input to the routine is always addressable in 24-bit mode. The following are 24-bit dependencies:

- The first GETMAIN should have LOC=BELOW because it has DCBs and OPEN and CLOSE short lists. Note that the OPEN and CLOSE parameter lists can be in storage above the line if MODE=31 is coded on the list and execute forms. LOC=ABOVE may be added to other GETMAINs.
- SYNAD and EODAD must reside below the line by having RMODE 24 for the CSECTs or there must be a DCBE with SYNAD and EODAD.
- Optionally add RMODE 31=BUFF to a DCBE to get QSAM buffers above the line. If so, you can delete the FREEPOOL macro.

Registers on Entry to the Auto-Delete Exit Routine

The following information is found in the registers on entry to the auto-delete exit routine.

Register	Contents
0	Unpredictable, must be saved and restored
1	Address of the auto-delete parameter list (mapped by macro CBRADUP)
2-12	Unpredictable, must be saved and restored
13	Address of a standard 18-word save area
14	Addressing mode and return address
15	Address of the auto-delete installation exit

Auto-Delete Installation Exit Parameter List

The auto-delete installation exit parameter list contains a user field (ADUUFLD). This field is binary zero on the first call to the installation exit. It is not changed by the system on any subsequent calls; therefore, the auto-delete installation exit can use it to save pertinent information. For example: the reentrant requirement makes it necessary to obtain (GETMAIN) virtual storage for save areas and temporary values. If your exit needs a dynamic work area, you can use GETMAIN once and store the address of the work area in ADUUFLD. This allows the exit to use the same storage area on subsequent calls in the cycle. Function code 2 (ADUDONE) is placed in ADUFUNC on the last call in the cycle. Your exit should check ADUFUNC and at end-of-cycle free up the dynamic work area space and any other resources it used.

Register 1 contains the address of the input parameter list for the auto-delete installation exit. The CBRADUP macro maps the parameter list, which contains the fields in Table 74.

Table 74. Auto–Delete Parameter List, CBRADUP

Offset	Type	Length or Bit Pattern	Name	Description
00 (X'00')	CHARACTER	160	CBRADUP	Class selection parm list
00 (X'00')	CHARACTER	16	ADUHDR	Header section
00 (X'00')	CHARACTER	4	ADUID	Block ID ‘ADU’
	CHARACTER ADU		ADUIDV	Control block ID
04 (X'04')	FIXED	4	ADULEN	Length of parameter list
08 (X'08')	FIXED	1	ADUVER	Version number
	DECIMAL 1		ADUVERV	Version number
09 (X'09')	FIXED	1	ADUREV	Revision number
	DECIMAL 0		ADUREVV	Revision number
10 (X'0A')	FIXED	1	ADUSP	Subpool number
	DECIMAL 0		ADUSPV	Subpool number
11 (X'0B')	FIXED	1	*	Reserved
12 (X'0C')	FIXED	4	*	Reserved
16 (X'10')	CHARACTER	8	ADULINK	Queue linkage section
16 (X'10')	ADDRESS	4	*	Reserved for compatibility Must contain zero
20 (X'14')	ADDRESS	4	*	Reserved for compatibility. Must contain zero
24 (X'18')	FIXED	1	ADUFUNC	Function code
	DECIMAL 1		ADUNOTFY	Notify delete call
	DECIMAL 2		ADUDONE	End of auto delete cycle call
25 (X'19')	BITSTRING	3	*	Reserved
28 (X'1C')	CHARACTER	44	ADUONAME	The object name

Table 74. Auto–Delete Parameter List, CBRADUP (continued)

Offset	Type	Length or Bit Pattern	Name	Description
72 (X'48')	CHARACTER	1	ADUBLANK	Separator blank
73 (X'49')	CHARACTER	44	ADUCLNAM	Collection name
117 (X'75')	CHARACTER	8	ADUCDAT	Object creation date
125 (X'7D')	CHARACTER	1	*	Reserved
126 (X'7E')	DECIMAL	2	ADUSCLEN	Length of storage class name
128 (X'80')	CHARACTER	30	ADUSCNAM	Name of the storage class to which object belongs
158 (X'9E')	DECIMAL	2	ADUMCLEN	Length of management class name
160 (X'A0')	CHARACTER	30	ADUMCNAM	Name of the management class to which object belongs
190 (X'BE')	DECIMAL	4	ADUSGLEN	Length of storage group name
194 (X'C2')	CHARACTER	8	ADUSGNAM	Name of the storage group to which object belongs
202 (X'CA')	CHARACTER	12	*	Reserved
214 (X'D6')	CHARACTER	32	ADUUFLD	User field
246 (X'F6')	CHARACTER	16	*	Reserved

Registers on Return from the CBRHADUX Installation Exit

The primary output from the installation exit is a return code in register 15 upon return to OSMC (see return code definitions listed in the input parameter list).

Other output includes information necessary to notify other applications when objects are deleted. Table 75 describes the return codes returned from the auto-delete Installation exit.

Table 75. Auto-Delete Return Codes, CBRADUP

Return Code	Name	Description
DECIMAL 0	ADUDELOK	Exit OKs object deletion.
DECIMAL 4	ADUNODEL	Exit rejects object deletion.
DECIMAL 8	ADUFAILC	Exit fails and should not be called again for this cycle deletions continue.
DECIMAL 12	ADUFAILN	Exit fails and should not be called again for this cycle. Deletions do not continue.

CBRHADUX Return and Reason Codes

00 (X'00') `IT IS OK TO DELETE THE OBJECT.`

04 (X'04') `DO NOT DELETE THE OBJECT.`

08 (X'08') `DELETE ALL OBJECTS WITHIN THIS GROUP WITHOUT CALLING THIS EXIT.`

12 (X'12') `DO NOT DELETE ANY OBJECTS WITHIN THIS GROUP AND DO NOT CALL THIS EXIT AGAIN.`

Sample Auto-Delete Installation Exit

SYS1.SAMPLIB member SAMPADUX, as shown in Figure 104 on page 513, is the sample auto-delete installation exit.

```
**********************************************************************
*                                                                    *
* $MOD(CBRHADUX),COMP(OSMC),PROD(OAM):                               *
*                                                                    *
*  MODULE NAME:  CBRHADUX                                            *
*                                                                    *
*                                                                    *
*  DESCRIPTIVE NAME:  OSMC SAMPLE AUTO-DELETE INSTALLATION EXIT      *
*                                                                    *
*                                                                    *
*PROPRIETARY V3 STATEMENT                                            *
*LICENSED MATERIALS - PROPERTY OF IBM                                *
*5695-DF1                                                            *
*(C) COPYRIGHT 1994  IBM CORP.                                       *
*END PROPRIETARY V3 STATEMENT                                        *
*                                                                    *
**********************************************************************
**********************************************************************
*                                                                    *
*  NOTE:                                                             *
*                                                                    *
*  THIS SAMPLE PROGRAM, IF INSTALLED AS IS, WILL PREVENT OBJECTS     *
*  FROM BEING DELETED.                                               *
*                                                                    *
*  RETURNS:       12 - DO NOT DELETE ANY OBJECTS WITHIN THIS GROUP   *
*                      AND DO NOT CALL THIS EXIT AGAIN FOR THIS      *
*                      GROUP                                         *
*                                                                    *
*  ADDITIONAL SAMPLE CBRHADUX FUNCTION HAS BEEN PROVIDED.            *
*  TO ENABLE MORE FUNCTION, CBRHADUX WILL HAVE TO BE ALTERED.        *
*  TWO LINES OF CODE FOLLOWING THE STANDARD ENTRY CODE BYPASS        *
*  THE FUNCTION DESCRIBED BELOW IN THIS PROLOGUE.                    *
*                                                                    *
**********************************************************************
**********************************************************************
*  FUNCTION:    VERIFY THE AUTOMATIC DELETION OF AN OBJECT. THE      *
*               VERIFICATION IS ACCOMPLISHED BY FIRST COMPARING      *
*               THIS OBJECT'S MANAGEMENT CLASS NAME WITH A           *
*               TABLE OF MANAGEMENT CLASS NAMES WHOSE OBJECTS,       *
*               RESIDING IN THE MANAGEMENT CLASS, SHOULD NOT         *
*               BE AUTOMATICALLY DELETED. THE VERIFICATION CONTINUES *
*               BY READING A SEQUENTIAL DATA SET THAT CONTAINS       *
*               THE NAMES OF OBJECTS THAT SHOULD NOT BE AUTO-        *
*               MATICALLY DELETED.  AN OBJECT'S COLLECTION NAME      *
*               IS STORED WITH OBJECT NAME IN THIS SEQUENTIAL        *
*               DATA SET TO DIFFERENTIATE BETWEEN OBJECTS WITH       *
*               SAME NAME BUT IN DIFFERENT COLLECTIONS.              *
*               RECORDS ARE ADDED TO THIS DATA SET WHEN OBJECTS      *
*               NEED TO BE KEPT LONGER THAN NORMAL.  WHEN THE        *
*               OBJECT IS NO LONGER NEEDED, ITS RECORD CAN BE        *
*               REMOVED FROM THE DATA SET.  IF THE DATA SET DOES     *
*               NOT EXIST (I.E., IS NOT CATALOGED), IT IS ASSUMED    *
*               THERE ARE NO OBJECTS THAT NEED VERIFICATION          *
*               AND THE PROCESS IS SKIPPED.                          *
*                                                                    *
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 1 of 14)

```
*              NOTIFY THAT AN OBJECT HAS BEEN DELETED. THE          *
*              NOTIFICATION IS ACCOMPLISHED BY ADDING AN OBJECT'S   *
*              NAME ALONG WITH ITS COLLECTION NAME TO A SEQUENTIAL  *
*              DATA SET.  THIS DATA SET SHOULD BE USED BY THE       *
*              INSTALLATION TO SYNCHRONIZE ITS DATA BASE WITH THE   *
*              OBJECT DIRECTORY (I.E., IF AN OBJECT IS DELETED FROM *
*              MANAGED STORAGE IT SHOULD ALSO BE DELETED FROM THE   *
*              APPLICATION FOLDER MANAGEMENT DATA BASE).            *
*                                                                   *
*   OPERATION: (COMPARE THE INPUT MANAGEMENT CLASS WITH THE TABLE   *
*              MANAGEMENT CLASSES WHOSE OBJECTS ARE NOT TO BE       *
*              DELETED--OPTIONAL CODE.)                             *
*              DYNAMICALLY ALLOCATE AND OPEN A SEQUENTIAL DATA SET  *
*              AND READ ALL OF THE NAMES TO BE VERIFIED INTO MAIN   *
*              STORAGE. THE OBJECTS ARE ONLY READ ONCE, THE FIRST   *
*              CALL. THE AREA REMAINS IN STORAGE FOR THE DURATION   *
*              OF THE OSMC CYCLE (FROM FIRST CALL TO END OF CYCLE   *
*              CALL.)                                               *
*              ALLOCATE AN OUTPUT DATA SET WITH A STATUS OF MOD     *
*              ALLOWING FOR THE ADDITION OF NAMES TO THE DATA SET.  *
*              OPEN THE OUTPUT DCB. THE OUTPUT DATA SET CONTAINS    *
*              ALL OF THE NAMES OF ALL OBJECTS (AND THEIR CORRES-   *
*              PONDING COLLECTION NAMES) THAT ARE DELETED.  THE     *
*              DATA SET REMAINS OPEN FOR THE DURATION OF THE OSMC   *
*              CYCLE.                                               *
*              COMPARE EACH INPUT OBJECT NAME AGAINST THOSE IN THE  *
*              MAIN STORAGE TABLE, IF A MATCH, CHECK TO SEE IF IN   *
*              THE SAME COLLECTION.  IF SAME COLLECTION, REJECT     *
*              THE DELETE.  IF NO MATCH, ADD THE OBJECT NAME ALONG  *
*              WITH ITS COLLECTION NAME TO THE OUTPUT DATA SET.     *
*              WHEN THE LAST CALL FUNCTION IS SPECIFIED, FREE THE   *
*              IN-STORAGE TABLE AND CLOSE THE OUTPUT DATA SET.      *
*              NAMES ARE COMPARED IN THEIR ENTIRETY. THAT IS THE    *
*              FULLY QUALIFIED NAME IS INPUT, ONE WHICH CONTAINS     *
*              THE OMDS SUFFIX, AND THEREFORE THE INPUT(VERIFY)     *
*              DATA SET MUST ALSO CONTAIN THE OBJECT NAME WITH THAT *
*              SUFFIX. SIMILARLY, THE OUTPUT (NOTIFY) DATA SET WILL *
*              CONTAIN THE FULLY QUALIFIED NAME.                    *
*                                                                   *
*  NOTES:                                                           *
*                                                                   *
*    DEPENDENCIES: THIS ROUTINE IS CALLED ON A GROUP BASIS. THAT    *
*                  IS, THE FIRST AND EVERY CALL OF A SINGLE         *
*                  INSTANCE OF THIS ROUTINE IS FOR THE SAME GROUP.  *
*                  THIS ALLOWS: A RETURN CODE INDICATING THE        *
*                  ROUTINE IS NOT INTERESTED IN A CERTAIN GROUP     *
*                  (E.G., 100 PEL GROUPS) AND THAT THE OUTPUT DATA  *
*                  SET(S) CAN BE SEGREGATED BY GROUP ALLOWING FOR   *
*                  PROCESSING OVERLAP (I.E., MORE THAN ONE TASK     *
*                  CAN HAVE AN INSTANCE OF THIS ROUTINE BECAUSE     *
*                  EACH TASK IS WRITING TO A SEPARATE DATA SET).    *
*                                                                   *
*                  THE END OF CYCLE CALL DOES NOT INCLUDE THE NAME  *
*                  OF AN OBJECT TO BE DELETED.                      *
*                                                                   *
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 2 of 14)

```
*                                                                     *
*     CHARACTER CODE: EBCDIC                                          *
*                                                                     *
*     RESTRICTIONS: NONE                                              *
*                                                                     *
*     REGISTER CONVENTIONS:                                           *
*                                                                     *
*               STANDARD ENTRY LINKAGE                                *
*                                                                     *
*               R0 AND R1 USED FOR SYSTEM SERVICE INVOCATION          *
*               R2 AND R3 ARE WORK REGISTERS                          *
*               R4  CONTAINS THE ADU ADDRESS                          *
*               R5 THROUGH R10 ARE WORK REGISTERS                     *
*               R11 CONTAINS THE ADDRESS OF THE DYNAMIC AREA          *
*               R12 IS THE BASE REGISTER                              *
*                                                                     *
*                                                                     *
*  INPUT:       THE ADU.                                              *
*                                                                     *
*               A SEQUENTIAL DATA SET CONTAINING OBJECT NAMES THAT    *
*               SHOULD NOT BE DELETED. THE DATA SET CONTAINS 89-BYTE  *
*               LOGICAL RECORDS EACH OF WHICH CONTAINS A SINGLE       *
*               OBJECT NAME, A SEPARATING BLANK, AND OBJECT'S         *
*               COLLECTION NAME.  THE NAMING CONVENTION FOR THIS      *
*               DATA SET:                                             *
*                PCATALOG.OBJECT.DELETE.VERIFY                        *
*                (WHERE: PCATALOG IS THE NAME OF A PRIVATE CATALOG)   *
*               THE ATTRIBUTES OF THE DATA SET ARE: FIXED BLOCKED,     *
*               LOGICAL RECORD LENGTH = 89 AND BLOCK SIZE = 8900.     *
*               BECAUSE THE LOGICAL RECORD LENGTH IS 89 AND IF        *
*               SEQUENCING IS USED, SEQUENCE NUMBERS WILL START       *
*               AT COLUMN 83 THRU 89 AND THEREFORE, RESULTS WILL      *
*               BE INVALID. THERE SHOULD NOT BE ANY SEQUENCE NUMBER   *
*               IN THE INPUT DATA SET.                                *
*                                                                     *
*  OUTPUT:      THE ADUUFLD CONTAINS THE ADDRESS OF THE DYNAMIC AREA  *
*               OBTAINED AT THE FIRST CALL. THIS AREA CONTAINS THE    *
*               ADDRESS OF THE MAIN STORAGE VERIFY TABLE AND THE      *
*               OUTPUT DCB THAT REMAINS OPEN ACROSS CALLS.            *
*                                                                     *
*               A SEQUENTIAL DATA SET CONTAINING 89 BYTE RECORDS.     *
*               EACH RECORD CONTAINS THE FULLY QUALIFIED NAME (THE    *
*               OBJECT NAME INCLUDES THE OMDS SUFFIX) OF THE OBJECT   *
*               THAT HAS BEEN DELETED. FULLY QUALIFIED COLLECTION     *
*               NAME FOLLOWS THE OBJECT NAME USING A BLANK AS A       *
*               SEPARATOR.  THE NAMING CONVENTION FOR DATA SET:       *
*                PCATALOG.SGRPNAME.OBJECT.DELETE.NOTIFY               *
*                (WHERE: PCATALOG IS THE NAME OF A PRIVATE CATALOG    *
*                        SGRPNAME IS THE NAME OF THE STORAGE GROUP *
*                        BEING PROCESSED)                             *
*   NOTE:  IF THIS DATASET IS PREALLOCATED PRIOR TO THE INVOCATION    *
*          OF THIS ROUTINE, THE DCB DECLARES OF THE DATASET SHOULD    *
*          BE REMOVED FROM THIS ROUTINE.  OTHERWISE, THE ALLOCATION   *
*          HERE WILL OVERRIDE THE PREALLOCATION OF THE DATASET,       *
*          CAUSING UNEXPECTED OUTPUT.                                 *
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 3 of 14)

```
*   NOTE:  PRIOR TO RUNNING THIS JOB,  YOU MUST MODIFY THE FOLLOWING *
*          STATEMENTS TO YOUR PRIVATE CATALOG AND STORAGE GROUP NAME.*
*          FIND THE FOLLOWING STATEMENTS IN THE CODE AND CHANGE      *
*          THE PCATALOG AND THE XXXXXXXX TO YOUR INSTALLATIONS NAMES:*
*                 1)  D1                                             *
*                 2)  NTFYDSN                                        *
*                 3)  NTFYDS2                                        *
*                                                                    *
*                                                                    *
*    RETURN CODES = 0  IT IS OKAY TO DELETE THE OBJECT               *
*                   4  DO NOT DELETE THE OBJECT                      *
*                   8  DELETE ALL OBJECTS WITHIN THIS GROUP WITHOUT  *
*                      CALLING THIS EXIT                             *
*                   12 DO NOT DELETE ANY OBJECTS WITHIN THIS GROUP   *
*                      AND DO NOT CALL THIS EXIT AGAIN               *
*                                                                    *
* $L0=OAM,110,082687,TUCWV:  INITIAL RELEASE                         *
* $L1=JDP3227,320,890523,TUCHTT:  RELEASE 1                          *
* $D1=JDP3227,320,890523,TUCLJS:  COLLECTION NAMES                   *
* $L2=PRESCOTT,331,901112,TUCLJS:  PRESCOTT SUPPORT                  *
**********************************************************************
         EJECT
CBRHADUX CSECT
* CBRHADUX  AMODE 31
* CBRHADUX  RMODE ANY
ADUX     DS    0H                  ENTRY POINT
         USING *,R15
         B     PASTID              BRANCH AROUND ID
         DC    CL8'CBRHADUX'
         DC    CL8'&
SYSDATE'
PASTID   DS    0H
         STM   R14,R12,12(R13)     SAVE CALLERS REGS
         LR    R12,R15             SET BASE REG
         DROP  R15
         USING ADUX,R12            ADDRESSIBILITY
         LR    R4,R1               GET INPUT PARAMETER LIST
         USING CBRADUP,R4
**********************************************************************
*                                                                    *
*    DO NOT DELETE OBJECTS IN THIS STORAGE GROUP AND DO NOT CALL     *
*               CBRHADUX AGAIN FOR THIS STORAGE GROUP                *
*                  -- TWO LINES OF BYPASS CODE --                    *
*                                                                    *
**********************************************************************
*                                                                    *
         LA    R2,ADUFAILN         LOAD RETURN CODE 12 INTO R2
         B     NOR13               BYPASS ALL CODE EXCEPT EXIT CODE
*                                                                    *
**********************************************************************
         L     R11,ADUUFLD         GET ADDRESS OF DYNAMIC AREA
         LTR   R11,R11             IS THIS THE FIRST CALL
         BNZ   NOGMAIN             NO, DO NOT GETMAIN
         GETMAIN RU,LV=DATALEN,SP=0  GET DYNAMIC AREA
         LR    R11,R1              GET STORAGE ADDRESS FOR DYNAMIC AREA
         USING DATAREA,R11         SET DYNAMIC AREA ADDRESSIBILITY FOR
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 4 of 14)

```
*                                REENTRANCY
         SLR   R1,R1             GET A ZERO
         ST    R1,RETCODE        ZERO THE RETURN CODE
         ST    R1,VRECSPTR       ZERO THE IN-STORAGE NAMES FIELD
         STC   R1,FLAGS          ZERO THE FLAGS BYTE
         ST    R11,ADUUFLD       SAVE DYNAMIC AREA FOR NEXT CALL
NOGMAIN  DS    0H
         ST    R13,SAVE1+4       BACKCHAIN SAVE AREAS
         LA    R2,SAVE1          GET SAVE AREA ADDRESS
         ST    R2,8(,R13)        FORWARDCHAIN SAVE AREAS
         LR    R13,R2            ESTABLISH SAVE AREA ADDRESS
         SR    R2,R2             CLEAR FOR ZERO
         ST    R2,RETCODE        ZERO THE INTERNAL RETURN CODE
         CLC   ADUSGNAM,=CL8'GROUP00' DO NOT PROCESS FURTHER IF
         BNE   NOTGRP00          100 PEL GROUP BEING PROCESSED
         LA    R2,ADUFAILC       INDICATE DELETES ARE OKAY BUT
         ST    R2,RETCODE        DO NOT CALL ADUX AGAIN
         B     ENDADUX           FOR THIS CYCLE
NOTGRP00 DS    0H
         CLI   ADUFUNC,ADUDONE   IS THIS AN END OF CYCLE CALL
         BE    DOFREE            YES, GO CLEANUP
         TM    FLAGS,TABUILT     HAS VERIFY TABLE BEEN BUILT
         BO    DOVERIFY          YES, DO NOT RE-OPEN DATA SET
         BAL   R14,OPENDS        FIRST CALL OPEN INPUT AND OUTPUT
         OI    FLAGS,TABUILT     INDICATE TABLE BUILT FOR NEXT CALL
*                                VRECSPTR WILL BE ZERO IF NO
*                                ENTRIES IN TABLE
DOVERIFY DS    0H
         L     R2,RETCODE        GET RETURN CODE
         LTR   R2,R2             DO NOT CONTINUE IF NON-ZERO
         BNZ   ENDADUX           END ADUX IF NON-ZERO CODE
         L     R10,VRECSPTR      GET ADDRESS OF IN-CORE VERIFY RECS
         LTR   R10,R10           IS THERE ANY VERIFY TABLE
         BZ    NOVERIFY          IF ZERO NO VERIFY NEEDED
         BAL   R14,VERIFY        CALL VERIFY ROUTINE
NOVERIFY DS    0H                FREEMAIN TABLE
         L     R2,RETCODE        GET THE RETURN CODE
         LTR   R2,R2             IF NOT ZERO VERIFY FAILED
         BNZ   NONOTIFY          NO NOTIFY IF VERIFY FAILED
         BAL   R14,NOTIFY        ADD OBJECT TO NOTIFY DATA SET
NONOTIFY DS    0H                DO NOT NOTIFY IF NON-ZERO CODE
ENDADUX  DS    0H                END OF CBRADUX
         LA    R3,ADUFAILC       RETURN CODE INDICATING NO
*                                RECALL OF ADUX FOR
*                                THIS CYCLE
         C     R3,RETCODE        IF ADUX IS TO BE CALLED AGAIN
         BH    NOFREE            DO NOT FREE AND CLOSE
DOFREE   DS    0H                FREE THE VERIFY TABLE AND CLOSE
*                                THE OUTPUT DATA SET
         BAL   R14,FREETAB       FREEMAIN THE VERIFY TABLE
         BAL   R14,CLOSEDS       CLOSE AND FREE OUTPUT DATA SET
         L     R13,SAVE1+4       GET CALLERS SAVE AREA ADDRESS
         L     R2,RETCODE        GET RETURN CODE
         SLR   R1,R1             GET A ZERO
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 5 of 14)

```
         ST    R1,ADUUFLD          CLEAR THE USER FIELD TO AVOID
*                                  INADVERTANT USE OF FREEMAINED
*                                  STORAGE
         FREEMAIN RU,LV=DATALEN,SP=0,A=(R11)  FREE DYNAMIC AREA
         B     NOR13               BYPASS R13 RESTORE
NOFREE   DS    0H                  BYPASS FREEMAIN
         L     R13,4(R13)          RESTORE CALLERS SAVE AREA
NOR13    DS    0H                  BYPASS R13 RESTORE IF FREEMAIN PATH
         L     R14,12(R13)         GET RETURN ADDRESS
         LR    R15,R2              GET RETURN CODE
         LM    R0,R12,20(R13)      RESTORE CALLERS REGS
         BSM   0,R14               RETURN TO CALLER
         EJECT
MCVERIFY DS    0H                  MC VERIFY SUBROUTINE
***********************************************************************
* MC VERIFY SUBROUTINE                                                *
*           SEARCH THE TABLE TO DETERMINE IF THIS OBJECT CAN BE       *
*           DELETED. IF THE INPUT MANAGEMENT CLASS IS LISTED IN       *
*           THE TABLE THEN NO OBJECTS WITH THIS MANAGEMENT CLASS      *
*           ARE TO BE DELETED.                                        *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         LA    R11,MCCNT           LOAD NUMBER OF TABLE ENTRIES
         LA    R10,MCTAB           LOAD ADDRESS OF TABLE IN R10
         USING TAB,R10             USE DSECT
COMPMC   CLC   MCNAME,ADUMCNAM     COMPARE MC NAMES
         BE    VMCMATCH
         LA    R10,TABLEN(,R10)
         BCT   R11,COMPMC
VMCMATCH DS    0H                  MC NAME IS IN VERIFY TABLE
         LA    R1,ADUNODEL         GOOD COMPARE, DO NOT DELETE
         ST    R1,RETCODE          SET DO NOT DELETE RETURN CODE
ENDMC    DS    0H                  END MC VERIFY SUBROUTINE
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
OPENDS   DS    0H                  OPENDS SUBROUTINE
***********************************************************************
*                                                                     *
* OPENDS SUBROUTINE                                                   *
*                                                                     *
*           OPEN THE INPUT DATA SET THAT CONTAINS THE NAMES OF        *
*           OBJECTS THAT SHOULD NOT BE DELETED.                       *
*                                                                     *
*           BUILD A TABLE CONTAINING THOSE NAMES. THE TABLE POINTER   *
*           IS SAVE SO THIS PROCESSING ONLY HAPPENS ONCE PER CYCLE.   *
*           THE TABLE IS A CHAINED SET OF 4K BLOCKS THAT. EACH BLOCK  *
*           CONTAINS 44 CHARACTER OBJECT NAMES IN SEQUENCE ALONG WITH *
*           44 CHARACTER COLLECTION NAME.  THE FIRST WORD IN THE      *
*           BLOCK CONTAINS A COUNTER OF THE NUMBER OF ENTRIES IN THE  *
*           BLOCK AND THE SECOND WORD CONTAINS A POINTER TO THE NEXT  *
*           BLOCK IF ANY.  A NEXT BLOCK ADDRESS OF ZERO INDICATES THE *
*           END OF CHAIN.                                             *
*                                                                     *
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 6 of 14)

```
*           OPEN THE OUTPUT DATA SET. THIS DATA SET WILL CONTAIN THE  *
*           NAMES OF ALL OBJECTS THAT WERE AUTOMATICALLY DELETED.     *
*                                                                     *
***********************************************************************
         SPACE 2
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         LA    R2,RENTAREA         GET TARGET ADDRESS FOR MOVE
         LA    R3,MOVELN           GET THE LENGTH OF THE MOVE
         LA    R6,STATAREA         GET SOURCE ADDRESS OF THE MOVE
         LR    R7,R3               GET LENGTH OF THE MOVE AND
*                                  PAD WITH ZEROS
         MVCL  R2,R6               COPY CONTROL BLOCKS TO
*                                  DYNAMIC AREA FROM STATIC AREA
         LA    R2,DYNRB            GET DYNAMIC RB ADDRESS
         ST    R2,S99RBPTR         STORE IN DYNAMIC RB POINTER FIELD
         OI    S99RBPTR,X'80'      SET END OF LIST FLAG
         LA    R2,TXTPTRV          GET DYNAMIC VERSION OF TXTPTR
         ST    R2,TXTPTR           STORE IN SVC 99 RB DYNAMIC VERSION
         LA    R2,0                GET ZERO FOR DDNAME TEXT UNIT PTR
         ST    R2,TXTDDN           PUT IT IN THE TXT UNIT PTR
         MVC   RETTXT(6),RETSTAT   MOVE FROM STATIC TO DYNAMIC
         LA    R2,RETTXT           LET SYSTEM DETERMINE DDNAME
         ST    R2,TXTRET
         LA    R1,S99RBPTR         SETUP REG 1 FOR SVC 99 CALL
         DYNALLOC                  ALLOCATE INPUT DATA SET
         LA    R2,4                GET DYNAMIC ENVIRONMENT ERROR
         CR    R15,R2              WAS DYNALLOC OKAY
         BL    DOOPENI             YES, OPEN VERIFY DATA SET
         BE    CHKNODS             IF A 4 SEE IF NO DATA SET
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
CHKNODS  DS    0H                  ENVIRONMENT ERROR
         LH    R15,=X'1708'        GET NO CATALOGED DS FAIL CODE
         CH    R15,DYNRB+4         FAILED BECAUSE DS NOT CATALOGED
         BE    NOOPENI             SO PROCEED WITHOUT VERIFY
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
DOOPENI  DS    0H                  ALLOCATED, DO OPEN
         LA    R1,OPENL            GET OPEN LIST ADDRESS
         LA    R8,DCBI             GET INPUT DCB ADDRESS
         USING IHADCB,R8
         MVC   DCBDDNAM,RETTXT+6   MOVE SYSTEM GENERATED DDN INTO DCB
         OPEN  ((R8),INPUT),MF=(E,(1)) OPEN INPUT DCB
         TM    DCBOFLGS,DCBOFOPN   WAS DS OPENED PROPERLY
         BO    OPENI               BUILD VERIFY TABLE
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
OPENI    DS    0H
         GETMAIN RU,LV=4096,SP=0   GET BLOCK FROM SUBPOOL ZERO
         LR    R5,R1               GET THE VERIFY TABLE ADDRESS
         SLR   R6,R6               GET A ZERO
*                                  ALSO, USE AS A RECORD COUNTER
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 7 of 14)

```
         ST    R6,0(,R5)            INDICATE IN-CORE TABLE NOT YET SET
         ST    R6,4(,R5)            INDICATE IN-CORE TABLE NOT YET SET
         LA    R9,8(,R5)            GET PAST RECORD COUNTER
         ST    R5,VRECSPTR          SAVE TABLE ADDRESS
GETINPUT DS    0H
         LA    R1,DCBI              GET INPUT DCB ADDRESS
         LR    R0,R9                ADDRESS OF RECORD
         GET   (1),(0)              READ A RECORD
         L     R2,RETCODE           CHECK FOR SYNAD ENTRY
         LTR   R2,R2                IF NON-ZERO SYNAD ENTERED
         BNZ   REODAD               CLOSE DATA SET AND END
         LA    R9,89(R9)            GET TO NEXT SLOT IN VERIFY TBL
         LA    R6,1(0,R6)           INCREMENT NUMBER OF RECORDS
         ST    R6,0(0,R5)           UPDATE COUNTER IN BLOCK
         C     R6,MAXNAMES          SEE IF MAX NAMES IN TABLE YET
         BE    GNEXTBLK             IF SO, GET A NEW BLOCK
         B     GETINPUT             READ UNTIL EODOD
GNEXTBLK DS    0H
         GETMAIN RU,LV=4096,SP=0    GET BLOCK FROM SUBPOOL ZERO
         ST    R1,4(0,R5)           CHAIN TO CURRENT TABLE
         LR    R5,R1                GET THE VERIFY TABLE ADDRESS
         SLR   R6,R6                CLEAR TO ZERO
         ST    R6,0(,R5)            INDICATE IN-CORE TABLE NOT YET SET
         ST    R6,4(,R5)            INDICATE IN-CORE TABLE NOT YET SET
         LA    R9,8(,R5)            GET PAST RECORD COUNTER
         B     GETINPUT             READ UNTIL EODOD
REODAD   DS    0H                   END OF DATA ON READ
         LA    R2,DCBI              GET INPUT DCB POINTER
         LA    R1,OPENL             GET AREA FOR CLOSE LIST
         CLOSE ((R2)),MF=(E,(1))    CLOSE INPUT DCB
         FREEPOOL DCBI              RELEASE BUFFER POOL
NOOPENI  DS    0H                   NO VERIFY DATA SET
         L     R2,RETCODE           CHECK FOR ERROR
         LTR   R2,R2                IF NON-ZERO ERROR OCCURRED
         BNE   ENDOPEN              END PROCESSING
         LA    R6,NTFYDS2
         L     R7,ADUSGLEN
         LA    R8,ADUSGNAM
         L     R9,ADUSGLEN
         MVCL  R6,R8
*
* UPDATE TEXT UNIT POINTERS FOR THE NOTIFY DATA SET
*
         LA    R3,0                 GET ZERO FOR DDNAME TEXT UNIT PTR
         ST    R3,TXTDDN            PUT IT IN THE TXT UNIT PTR
         LA    R3,DSNTXTN           GET NOTIFY DSNAME TEXT UNIT PTR
         ST    R3,TXTDSN            PUT IT IN THE TXT UNIT PTR
*                                   INSTEAD OF VERIFY DSNAME
         LA    R3,MODTXT            GET STAT=SHR TEXT UNIT PTR
         ST    R3,TXTSTAT           STORE NEW TEXT UNIT PTR
         LA    R3,CTLGTXT           GET DISP=,KEEP TEXT UNIT PTR
         ST    R3,TXTDISP           STORE NEW TEXT UNIT PTR
         LA    R3,TRKTXT            GET TRACK ALLOATION TU PTR
         ST    R3,TXTTRK            STORE NEW TEXT UNIT PTR
         LA    R3,PRIMTXT           GET PRIMARY AMOUNT TU PTR
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 8 of 14)

```
         ST    R3,TXTPRIM          STORE NEW TEXT UNIT PTR
         LA    R3,SECTXT           GETSECONDARY AMOUNT TU PTR
         ST    R3,TXTSEC           STORE NEW TEXT UNIT PTR
         MVC   RETTXT(6),RETSTAT   MOVE FROM STATIC TO DYNAMIC
         LA    R3,RETTXT           LET SYSTEM DETERMINE DDNAME
         ST    R3,TXTRET
         LA    R1,S99RBPTR         SET UP FOR SVC 99
         DYNALLOC                  FREE THE DATA SET
         LTR   R15,R15             IF DYNALLOC OKAY
         BZ    OPENO               OPEN NOTIFY DATA SET
         LA    R15,ADUFAILN        ERROR CAUSES NO RETURN TO ADUX
         ST    R15,RETCODE         SET RETURN CODE
         B     ENDOPEN             END ADUX
OPENO    DS    0H                  OPEN OUTPUT DATA SET
         LA    R1,OPENL            GET OPEN LIST ADDRESS
         LA    R8,DCBO             GET INPUT DCB ADDRESS
         MVC   DCBDDNAM,RETTXT+6   MOVE SYSTEM GENERATED DDN INTO DCB
         OPEN  ((R8),OUTPUT),MF=(E,(1)) OPEN OUTPUT DCB
         USING IHADCB,R8
         TM    DCBOFLGS,DCBOFOPN   WAS DS OPENED PROPERLY
         BNO   NOOPENO             SET NO NOTIFY TABLE
         OI    FLAGS,ODCBO         INDICATE NOTIFY DATA SET IS OPEN
         B     ENDOPEN             END DATA SET OPENING
NOOPENO  DS    0H                  OUTPUT DATA SET DID NOT OPEN
         LA    R2,ADUFAILN         SET FAILING RETURN CODE
         ST    R2,RETCODE          DO NOT RECALL ADUX
ENDOPEN  DS    0H                  END OF OPENDS SUBROUTINE
         L     R14,SAVE14          GET RETURN ADDRESS
         BR    R14                 RETURN TO MAINLINE
         EJECT
VERIFY   DS    0H                  VERIFY SUBROUTINE
***********************************************************************
* VERIFY SUBROUTINE                                                   *
*           SEARCH THE TABLE BUILT BY OPENDS. IF THE INPUT OBJECT      *
*           NAME IS IN THE TABLE, SEND A RETURN CODE INDICATING       *
*           THE AUTODELETE SHOULD NOT BE DONE.                         *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         L     R5,VRECSPTR         GET FIRST TABLE ADDRESS
COMPLOOP DS    0H                  COMPARE LOOP
         L     R3,0(,R5)           GET COUNT OF ENTRIES
         LA    R2,8(,R5)           GET ADDRESS OF FIRST NAME
         LTR   R3,R3               IF EMPTY FILE
         BZ    ENDVERF             GO TO ENDVERF
COMPNAME DS    0H                  NAME COMPARE LOOP
         CLC   0(44,R2),ADUONAME   COMPARE NAME IN VERIFY TABLE
*                                  AGAINST OBJECT BEING DELETED
         BE    VCOLNAM             IF MATCH, CHECK COLLECTION NAME
DIFCOLNM LA    R2,89(,R2)          GET ADDRESS OF NEXT NAME
         BCT   R3,COMPNAME         COMPARE NEXT NAME
         L     R5,4(,R5)           GET NEXT TABLE SECTION ADDRESS
         LTR   R5,R5               IS THERE IS A NEXT SECTION
         BZ    ENDVERF             END VERIFY IF NOT
         B     COMPLOOP            COMPARE NEXT NAME
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 9 of 14)

```
VCOLNAM  DS    0H                   OBJ MATCH, CHECK COL NAME
         CLC   45(44,R2),ADUCLNAM  OBJ MATCH, SAME COL NAME?
         BE    VMATCH               IF MATCH, DO NOT DELETE
         B     DIFCOLNM             NO MATCH, GET NEXT OBJ NAME
VMATCH   DS    0H                   OBJECT NAME IS IN VERIFY TABLE
         LA    R1,ADUNODEL          GOOD COMPARE, DO NOT DELETE
         ST    R1,RETCODE           SET DO NOT DELETE RETURN CODE
ENDVERF  DS    0H                   END VERIFY SUBROUTINE
         L     R14,SAVE14           GET SAVED R14
         BR    R14                  RETURN TO MAINLINE
         EJECT
NOTIFY   DS    0H                   NOTIFY SUBROUTINE
***********************************************************************
* NOTIFY SUBROUTINE                                                   *
*        WRITE THE OBJECT AND COLLECTION NAMES TO NOTIFY DATA SET.    *
*        THIS DATA SET WILL BE READ BY THE IMS SYSTEM LATER TO REMOVE *
*           THE OBJECT FROM THE IMS DATA BASE.                        *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14           SAVE RETURN ADDRESS
         LA    R1,DCBO              GET OUTPUT DCB ADDRESS
         LA    R0,ADUOBJCL          GET OBJECT NAME ADDRESS
         PUT   (1),(0)              WRITE NAME TO NOTIFY DATA SET
         L     R14,SAVE14           GET SAVED R14
         BR    R14                  RETURN TO MAINLINE
         EJECT
FREETAB  DS    0H                   FREEMAIN VERIFY TABLE SUBROUTINE
***********************************************************************
* FREETAB SUBROUTINE                                                  *
*           FREEMAIN THE IN-STORAGE VERIFY TABLE                      *
***********************************************************************
         SPACE 1
         ST    R14,SAVE14           SAVE RETURN ADDRESS
         L     R5,VRECSPTR          GET FIRST TABLE ADDRESS
FREELOOP DS    0H                   LOOP THROUGH CHAINED TABLES
         L     R3,4(,R5)            GET NEXT TABLE ADDRESS
         LTR   R5,R5                IS THERE A TABLE ADDRESS
         BZ    ENDFREE              IF NOT, END FREEMAIN LOOP
         FREEMAIN RU,LV=4096,A=(R5) FREE TABLE SECTION
         LR    R5,R3                ADDRESS NEXT SECTION TO FREE
         B     FREELOOP             FREEMAIN NEXT SECTION
ENDFREE  DS    0H                   END OF TABLE FREEMAIN LOOP
         L     R14,SAVE14           GET SAVED R14
         BR    R14                  RETURN TO MAINLINE
         EJECT
CLOSEDS  DS    0H                   CLOSE DATA SET SUBROUTINE
***********************************************************************
*                                                                     *
* CLOSEDS SUBROUTINE                                                  *
*                                                                     *
*           CLOSE THE OPEN NOTIFY DATA SET                            *
*                                                                     *
***********************************************************************
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 10 of 14)

```
         SPACE 1
         ST    R14,SAVE14          SAVE RETURN ADDRESS
         TM    FLAGS,ODCBO         CHECK TO SEE IF OUTPUT DS IS OPEN
         BNO   NOCLOSE             DO NOT ISSUE CLOSE IF NOT OPEN
         LA    R1,OPENL            GET AREA FOR CLOSE LIST
         LA    R2,DCBO             GET DCB ADDRESS
         CLOSE ((R2)),MF=(E,(1))   CLOSE OUTPUT DCB NOTIFY DSN
         FREEPOOL DCBO             RELEASE OUTPUT BUFFER POOL
NOCLOSE  DS    0H                  DO NOT CLOSE
         L     R14,SAVE14          GET SAVED R14
         BR    R14                 RETURN TO MAINLINE
         EJECT
ADUXSYN  DS    0H
***********************************************************************
*                                                                     *
* I/O ERROR SYNAD                                                     *
*                                                                     *
*      AN I/O ERROR DURING READ OR WRITE RESULTS IN SETTING THE       *
*      RETURN CODE INDICATING DO NOT DELETE AND DO NOT CALL           *
*      ADUX AGAIN FOR THIS CYCLE.                                     *
*                                                                     *
***********************************************************************
         SPACE 1
         LA    R2,ADUFAILN         SET FAIL CODE TO NO DELETE AND
         ST    R2,RETCODE          DO NOT CALL ADUX FOR THIS CYCLE
         BR    R14                 RETURN TO MAINLINE
         EJECT
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
MCTAB    DC    C'MCNODEL1'         MANAGEMENT CLASS TABLE
         DC    C'MCNODEL2'
MCCNT    EQU   (*-MCTAB)/8
MAXNAMES DC    F'92'               MAXIMUM NUMBER OF NAMES IN
*                                  A 4K BLOCK OF VERIFY TABLE
RETSTAT  DC    AL2(85)
         DC    AL2(1)
         DC    AL2(8)
DDNMTXT  DS    0F
         DC    AL2(1)
         DC    AL2(1)
         DC    AL2(ENDDDN-STRTDDN)
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 11 of 14)

```
STRTDDN  DC    C'CBRADUXI'
ENDDDN   EQU *
MODTXT   DC    AL2(4)              STATUS=MOD
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(2)
SHRTXT   DC    AL2(4)              STATUS=SHR
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(8)
TRKTXT   DC    AL2(7)              TRACK ALLOCATION IF NOT OLD
         DC    AL2(0)
PRIMTXT  DC    AL2(10)
         DC    AL2(1)
         DC    AL2(3)
         DC    AL3(10)             10 TRACKS PRIMARY ALLOCATION
SECTXT   DC    AL2(11)
         DC    AL2(1)
         DC    AL2(3)
         DC    AL3(10)             10 TRACKS SECONDARY ALLOCATION
CTLGTXT  DC    AL2(5)              DISP=CATLG
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(2)
KEEPTXT  DC    AL2(5)              DISP=KEEP
         DC    AL2(1)
         DC    AL2(1)
         DC    AL1(8)
UNITTXT  DC    AL2(21)
         DC    AL2(1)
         DC    AL2(5)
         DC    C'SYSDA'
CLOSTXT  DC    AL2(28)             FREE DATA SET WHEN CLOSED
         DC    AL2(0)
DSNTXT   DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDDSN-VRFYDSN)
VRFYDSN  DC    C'PCATALOG.OBJECT.DELETE.VERIFY'
ENDDSN   EQU   *
STATAREA DS    0F                  STATIC CBS TO BE MOVED
         OPEN   (,),MF=L
DCBSTAT  DS    0F
*   NOTE:  IF THIS DATASET IS PREALLOCATED PRIOR TO THE INVOCATION   *
*          OF THIS ROUTINE, THE DCB DECLARES OF THE DATASET SHOULD   *
*          BE REMOVED FROM THIS ROUTINE.  OTHERWISE, THE ALLOCATION  *
*          HERE WILL OVERRIDE THE PREALLOCATION OF THE DATASET,      *
*          CAUSING UNEXPECTED OUTPUT.                                *
*                                                                    *
         DCB  DDNAME=CBRADUXO,MACRF=(PM),OPTCD=W,LRECL=89,BLKSIZE=8900,
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN
         DCB  DDNAME=CBRADUXI,MACRF=(GM),LRECL=89,BLKSIZE=8900,
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN,EODAD=REODAD
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 12 of 14)

```
DYNRBC   DS    0F
         DC    AL1(20)              RB LENGTH
         DC    AL1(01)              DSNAME ALLOCATION
         DC    X'C0'                FLAGS1 - NO EXIST ALLOC
         DC    X'0' FLAGS1
         DC    F'0'                 ERROR CODES
         DC    A(0)                 TEXT UNIT POINTERS
         DC    F'0'                 RESERVED
         DC    X'00'                WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
*        DC    X'D1'                WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
         DC    AL3(0)
TXTPTRC  DS    0F
         DC    A(DSNTXT)
         DC    A(DDNMTXT)
         DC    A(SHRTXT)
         DC    A(KEEPTXT)
         DC    A(0)                 TRACK TEXT UNIT FOR OUTPUT DS
         DC    A(0)                 PRIMARY TEXT UNIT FOR OUTPUT DS
         DC    A(0)                 SECONDARY TEXT UNIT FOR OUTPUT DS
         DC    A(0)                 RETURN DDNAME FOR OUTPUT DS
         DC    A(UNITTXT)           UNIT TEXT UNIT FOR OUTOUT DS
         DC    A(CLOSTXT)           FREE (UN-ALLOCATE) AT CLOSE
         DC    X'80'
         DC    AL3(0)
         DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDD1-D1)
D1       DC    C'PCATALOG.XXXXXXXX.OBJECT.DELETE.NOTIFY'
ENDD1    EQU   *
MOVELN   EQU   *-STATAREA
         EJECT
         CBRADUP
         EJECT
***********************************************************************
*
*      MANAGEMENT CLASS DSECT TO MAP TO MCTAB
*
***********************************************************************
TAB      DSECT ,
MCLEN    DS    H
MCNAME   DS    CL30
TABLEN   EQU   *-TAB
DATAREA  DSECT
SAVE1    DS    18F                  SAVE AREA
SAVE14   DS    1F                   R14 SAVE AREA FOR SUBROUTINES
FLAGS    DS    X                    FLAG AREA
TABUILT  EQU   X'80'                VERIFY TABLE HAS BEEN BUILT
ODCBO    EQU   X'40'                NOTIFY DATA SET IS OPEN
VRECSPTR DS    F                    ADDRESS OF THE IST 4K BLOCK OF
RETTXT   DS    AL2
         DS    AL2
         DS    AL2
         DS    CL8
*                                   OBJECT NAMES READ FROM THE VERIFY
*                                   DATA SET
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 13 of 14)

```
RETCODE  DS    F                    INTERNAL RETURN CODE
S99RBPTR DS    F                    ADDRESS OF SVC99 RB
RENTAREA DS    0F                   REENTRANT COPY OF STATIC DEFINED
*                                   CONTROL BLOCKS
OPENL    OPEN   (,),MF=L
DCBS     DS    0F
DCBO     DCB  DDNAME=CBRADUXO,MACRF=(PM),OPTCD=W,LRECL=89,BLKSIZE=8900,
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN
DCBI     DCB  DDNAME=CBRADUXI,MACRF=(GM),LRECL=89,BLKSIZE=8900,
               DSORG=PS,RECFM=FB,SYNAD=ADUXSYN,EODAD=REODAD
DYNRB    DS    0F
         DC    AL1(20)              RB LENGTH
         DC    AL1(01)              DSNAME ALLOCATION
         DC    X'C0'                FLAGS1 - NO EXIST ALLOC
         DC    X'0' FLAGS1
         DC    F'0'                 ERROR CODES
TXTPTR   DC    A(TXTPTRV)           TEXT UNIT POINTERS
         DC    F'0'                 RESERVED
*        DC    X'D1'                WAIT FOR VOLS,UNITS,DSNS AND MOUNTS
         DC    X'00'
         DC    AL3(0)
TXTPTRV  DS    0F
TXTDSN   DC    A(DSNTXT)
TXTDDN   DC    A(DDNMTXT)
TXTSTAT  DC    A(SHRTXT)
TXTDISP  DC    A(KEEPTXT)
TXTTRK   DC    A(0)
TXTPRIM  DC    A(0)
TXTSEC   DC    A(0)
TXTRET   DC    A(0)
TXTUNIT  DC    A(UNITTXT)
         DC    A(CLOSTXT)
         DC    X'80'
         DC    AL3(0)
DSNTXTN  DC    AL2(2)
         DC    AL2(1)
         DC    AL2(ENDDSNN-NTFYDSN)
NTFYDSN  DC    C'PCATALOG.'
NTFYDS2  DC    C'XXXXXXXX.OBJECT.DELETE.NOTIFY'
ENDDSNN  EQU   *
DATALEN  EQU   *-DATAREA
         DCBD    DSORG=QS
         IEFZB4D2
         END   CBRHADUX
```

Figure 104. Sample OSMC Auto-Delete Installation Exit (Part 14 of 14)

Appendix F. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Information Enabling Requests
Dept. DZWA
5600 Cottle Road
San Jose, CA 95193 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Programming Interface Information

This book helps you plan, install, and administer the Object Access Method (OAM), a component of DFSMSdfp. This book primarily documents intended programming interfaces that allow the customer to write programs to obtain the services of the OAM component of DFSMSdfp.

This book also documents information that is NOT intended to be used as Programming Interfaces of the OAM component of DFSMSdfp. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

NOT Programming Interface information

End of NOT Programming Interface information

Trademarks

The following terms are trademarks of the IBM Corporation in the United States, or other countries, or both:

3090	BookManager	CICS
CICS/ESA	CICS/MVS	DATABASE 2
DB2	DFSMSdfp	DFSMSdss
DFSMShsm	DFSMSrmm	DFSMS
ES/3090	ESA/390	ESCON
GDDM	IBM	ImagePlus
IMS	IMS/ESA	MVS/DFP
MVS/ESA	MVS/SP	OS/2
OS/390	Parallel Sysplex	QMF
RACF	RMF	System/370
z/OS		

The following terms are trademarks of other companies:

FileNet	FileNet Corporation
OSAR	
LMSI	Laser Magnetic Storage International
LaserDrive	

Glossary

This glossary defines technical terms and abbreviations used in Object Access Method documentation. If you do not find the term you are looking for, refer to the index of the appropriate Object Access Method manual or view *IBM Dictionary of Computing*, located at:

http://www.ibm.com/networking/nsg/nsgmain.htm

This glossary may include terms and definitions from:

- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York 10036.
- *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Electrotechnical Commission (ISO/IEC JTC2/SC1).

Numerics

3480. IBM 3480 Magnetic Tape Subsystem. A group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A11 and A22, and two drive models, B11 and B22.

3490. IBM 3490 Magnetic Tape Subsystem. A group of magnetic tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A01 and A02, and two drive models, D31 and D32, in which the control unit function and tape drives are integrated.

3490E. IBM 3490E Magnetic Tape Subsystem. A group of enhanced capability tape controllers and drives supporting cartridge system tape (as opposed to reel tape). There are two controller models, A10 and A20, and two drive models, B10 and B20. In addition, there are two models, D41 and D42, in which the control unit function and tape drives are integrated.

3590. The IBM TotalStorage Enterprise High Performance Tape System 3590 is capable of coexisting with 3490 and 3490E devices in the IBM TotalStorage Enterprise Automated Tape Library (3494), or as a stand-alone tape drive. The 3590 has a built-in control unit. This device sypports the IBM High Performance Cartridge System Tape and the IBM Extended High Performance Cartridge System Tape media.

A

access path. The path DB2 uses to get to data specified in SQL statements. An access path can involve an index, a sequential search, or a combination of both.

ACDS. Active control data set.

ACS. Automatic class selection.

active configuration. The configuration currently used by SMS to control the managed storage in the installation.

active control data set (ACDS). A VSAM linear data set that contains a copy of an active configuration and subsequent updates. All systems in an SMS complex use the ACDS identified in the IGDSMS*xx* member of the PARMLIB to manage storage.

address space. One or more unique identifiers assigned to OAM and OTIS sessions.

aggregate group. Groups a collection of data objects that form a data type, which allows the data to be referred to collectively or individually. Aggregate group is used in conjunction with the storage of DASD data, not within an OAM environment.

alphanumeric. The set of characters that contains only the numbers 0–9 and the uppercase letters A–Z.

application plan. The control structure produced during the bind process and used by DB2 to process SQL statements during application execution.

ATLDS. Automated tape library dataserver.

attribute. A named property of an entity.

automated tape library dataserver (ATLDS). A device consisting of robotic components, cartridge storage frames, tape subsystems, and controlling hardware and software, together with the set of volumes that resides in the library and which may be mounted on the library tape drives.

automatic cartridge loader. An optional feature for the 3480 tape drive. It allows the automatic loading of tape cartridges that have been placed into a loading rack. Manual loading of single tape cartridges is also possible.

automatic class selection (ACS). Routines that determine the storage class, management class, and storage group for a collection. The storage administrator is responsible for establishing ACS routines appropriate to an installation's storage requirements.

B

backup. The first or second backup copy of a primary object in an Object Backup storage group. The only way to access this copy of the object is to recover the single object (single object recovery operator command) or the entire optical disk on which it was stored.

bind. The process by which the output from the DB2 precompiler is converted to a usable control structure called an application plan. This process is the one during which access paths to the data are selected and some authorization checking is performed.

block.

C

CAF. Call attachment facility.

call attachment facility (CAF). A DB2 attachment facility that allows application programs to connect to and use DB2.

cartridge. See *optical disk cartridge*.

cartridge accessor. See *robot*.

cartridge system tape. The standard tape cartridge medium, 160 meters in length.

CCW. Continuous composite WORM.

CDS. Control data set.

CDS base. Control data set base.

central processing unit (CPU). The circuitry of a computer that controls the interpretation and execution of instructions. Traditionally, the complete unit was often regarded as the CPU, whereas today the CPU is often a microchip. In either case, the centrality of a processor or processing unit depends on the configuration of the system or network in which it is used.

central processor. The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program, load, and other machine operations.

central processor complex. A physical connection of hardware that consists of main storage, one or more central processors, timers, and channels.

CFRM. Coupling facility resource management.

Channel-to-channel. A method of connecting two computing devices.

CICS. Customer Information Control System.

class transition. A change in an object's management class, storage class, or both. When an event occurs that brings about a change in an object's service level or management criteria. Class transition occurs during a storage management cycle.

collection. A group of objects that typically have similar performance, availability, backup, retention, and class transition characteristics. A collection is used to catalog a large number of objects which, if cataloged separately, could require an extremely large catalog.

COMMDS. Communications data set.

commit. In DB2, to cause all changes that have been made to the database file since the last commitment operation to become permanent, and the records to be unlocked so they are available to other users.

communications data set (COMMDS). The primary means of communication among systems in an SMS complex. Shared among the systems in the SMS complex, the COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume.

compaction. See *improved data recording capability*.

compatibility mode. Mode of running DFSMS/MVS 1.4.0 in which no more than eight unique systems or system group names are supported in the SMS configuration. When running in this mode, the DFSMS/MVS 1.4.0 system may share configurations and communications data sets with the systems running down level release of DFSMS/MVS or MVS/DFP. Otherwise, configurations and communications data sets may not be shared.

complex. See *SMS complex*.

connectivity. Relationship establishing the eligibility of a given system in an SMS complex to access a VIO, pool, object, object backup, or tape storage group, or the individual volumes within a pool storage group. The relationship can be NOTCON (not connected), indicating eligibility, or any of the following, all of which imply some level of eligibility: ENABLE, QUIALL (quiesce all), QUINEW (quiesce new, DISALL (disable new), DISNEW (disable new).

construct. One of the following: data class, management class, storage class, storage group, or aggregate group.

control data set (CDS). With respect to SMS, a VSAM linear data set containing configurational, operational, or communication information. SMS uses three types of control data sets: the source control data set (SCDS), the active control data set (ACDS), and the communications data set (COMMDS).

controlling library. A 3995 model 131, 132, 133, or C3A optical library model containing the control unit for the 3995 Optical Library Dataserver.

CONVERT. A physical volume status indicating all the data sets on a given volume have an associated storage class and are not cataloged in an integrated catalog facility catalog. SMS can select a CONVERT volume for all supported functions. CONVERT volumes are sometimes referred to as converted. See also *INITIAL*.

coupling facility. A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

CPC. Central processor complex.

CPU. Central processing unit.

Cross-system coupling facility (XCF). A component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.

CTC. Channel-to-channel.

D

DASD. Direct access storage device.

DATABASE 2 (DB2). A relational database management system.

DATABASE 2 interactive (DB2I). An interactive relational database management program.

Database request module. A data set member created by the DB2 precompiler that contains information about SQL statements. DBRMs are input into the bind process.

data class. A named template for data set allocation. Information supplied by data class may be SPACE, LRECL, BLKSIZE, RECFM, and others.

data compaction. See *improved data recording capability*.

Data Facility Product (DFP). An IBM licensed program used to manage programs, devices, and data in an MVS environment.

Data Facility Storage Management Subsystem (DFSMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Data Facility Storage Management Subsystem data facility product (DFSMSdfp). An IBM licensed program used to manage programs, devices, and data in an MVS environment. It is a component of the z/OS DFSMS system.

dataplex. A set of systems in a multisystem environment sharing system-managed data on system-managed storage devices.

DB2. DATABASE 2.

DB2 data sharing. The ability of concurrent DB2 subsystems or application programs to directly access and change the same data while maintaining data integrity.

DB2 data sharing group. A collection of one or more concurrent DB2 subsystems that directly access and change the same data while maintaining data integrity.

DB2I. DATABASE 2 interactive.

DBRM. Database request module.

device. This term is used interchangeably with unit. For a disk or tape, a unit on which a volume may be mounted. For example, a tape drive is a device; a tape cartridge is a volume. Device also applies to other types of equipment, such as a card reader or a channel-to-channel (CTC) adapter.

device group. A group of devices that are interchangeable as far as MVS allocation is concerned. Unless a request is for a specific device name, if one device in a given device group can satisfy a request, any other can also satisfy that request.

device name. This term is used interchangeably with device number, unit number, and unit name. It is the number by which a specific device is known. For example, and installation with two tape drives might assign them device names 181 and 182.

DFP. Data Facility Product.

DFSMS. Data Facility Storage Management Subsystem.

DFSMSdfp. Data Facility Storage Management Subsystem data facility product.

DFVT. Data Facilities Vector Table.

direct access storage device (DASD). A device in which time is effectively independent of the location of the data.

DISALL (disable all). Relationship preventing a system from allocating new data sets in a VIO, pool, object, object backup, or tape storage group, or on individual volumes within a pool storage group.

disk. See *optical disk*.

DIV. Data-in-Virtual.

E

Environmental Record Editing and Printing program (EREP). The program that formats and prepares reports from the data contained within the environmental recording data set (ERDS).

EPI. ERDS Physical Identifier.

EREP. Environmental Record Editing and Printing program.

esoteric unit name. An installation-assigned name for a set of devices intended to be used when requesting a device. For example, an esoteric unit name of TAPE might represent all tape devices in an installation.

external label. A machine- and human-readable label attached to the outside of a tape cartridge that is to be stored in a tape library. The label contains the volume serial number of the tape volume.

F

FSC. Fault Symptom Code.

G

GB. Gigabyte.

generalized trace facility (GTF). An optional OS/VS service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

gigabyte (GB). When referring to storage capacity, two to the thirtieth power; 1 073 741 824 in decimal notation.

global resource serialization (GRS). A function that provides an MVS serialization mechanism for resources (typically data sets) across MVS images.

GMT. Greenwich Mean Time.

grant. A DB2 process that authorizes users to access data.

GRS. Global resource serialization.

GTF. Generalized trace facility.

H

hardware configuration dialog. See *MVS/ISI*.

HCD. Hardware configuration dialog.

I

IARS. Initial Access Response Seconds.

ICF. Integrated catalog facility.

ICL. Integrated cartridge loader.

ID. Identification.

identification. In computer security, the process that allows a system to recognize an entity by means of personal, equipment, or organizational characteristics or codes.

IDRC. Improved data recording capability.

image copy. An exact reproduction of all or part of a table space. DB2 provides utilities to make full image copies or incremental image copies.

improved data recording capability (IDRC). A form of compression used when storing data on tape. This can increase the effective cartridge data capacity and the effective data transfer rate.

IMS. Information Management System.

index. A set of pointers that are logically ordered by the values of a key. Indexes provide quick access to data and can enforce uniqueness on the rows in a DB2 storage table.

Information Management System (IMS). A transaction and hierarchical database management system that organizes the data in different structures, depending on data type, to optimize storage and retrieval, and to ensure integrity and ease of recovery.

initial access response seconds (IARS). A parameter specified in the definition of an SMS storage class indicating the desired response time to locate, mount, and prepare a piece of media for data transfer.

initial program load (IPL). (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of leading system programs and preparing a system to run jobs. (4) Synonymous with system restart, system startup.

installation exit. The means specifically described in an IBM software product’s documentation by which an IBM software product may be modified by a customer’s system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software product.

integrated cartridge loader (ICL). A standard feature for the 3490 tape drive. It allows the automatic loading of tape cartridges that have been placed into a loading rack. Manual loading of single tape cartridges is also possible.

integrated catalog facility (ICF). In the Data Facility Product (DFP), a facility that provides for integrated catalog facility catalogs.

integrated catalog facility catalog. In the Data Facility Product (DFP), a catalog that consists of a basic catalog structure, which contains information about VSAM and non-VSAM data sets, and at least one VSAM volume data set, which contains information about VSAM data sets only.

Interactive Problem Control System (IPCS). A system facility that allows interactive problem analysis.

Interactive System Productivity Facility (ISPF). An interactive base for ISMF.

IPCS. Interactive Problem Control System.

IPL. Initial program load.

ISMF. Interactive Storage Management Facility.

ISO. International Organization for Standardization.

ISPF. Interactive System Productivity Facility.

K

KB. Kilobyte.

kilobyte (KB). As used in data processing (1 024 bytes).

KLOC. Thousands of lines of code.

L

LCS. Library Control System.

Library Control System (LCS). Component of OAM that writes and reads objects on optical disk storage, and manipulates the optical volumes on which the objects reside.

library expansion unit. A 3995 model 111, 112, 113, C12, C14, C16, C18, C32, C34, C36, or C38 that connects to a controlling library to expand the capacities of the 3995 Optical Library Dataserver.

M

magneto-optic (MO) recording. A method of storing information magnetically on optical media, which is sensitive only at high temperatures, while stable at normal temperatures. A laser is used to heat a small spot on the medium for recording. The ability to focus the laser tightly greatly increases the data density over standard magnetic media. MO media are erasable and rewritable.

management class. A named collection of management attributes describing the retention, backup, and class transition characteristics for a group of objects in an object storage hierarchy.

manual tape library (MTL). An installation-defined set of tape drives and the set of volumes that can be mounted on the drives.

max connects. The maximum amount of foreground and background users and TSO connections allowed to a DB2 subsystem.

MB. Megabyte.

megabyte (MB). As used in data processing (1 048 576 bytes).

MO. Magneto-optic recording technique for optical media.

mount. A host-initiated operation that results in a tape cartridge being physically inserted into a tape drive.

MTL. Manual tape library.

MVS Configuration Program (MVSCP). A program that defines the hardware I/O configuration to the MVS operating system.

MVSCP. MVS configuration program.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/ISI. A set of user-friendly dialogs for interacting with MVS. In particular, MVS/ISI HCD is the hardware configuration dialog. It is the replacement for MVSCP.

O

OAM. Object Access Method.

OAMplex. The concept of connecting instances of OAM to a single XCF (cross-system coupling facility) group to create an OAMplex within the parallel sysplex environment. This includes using DB2 data sharing where the scope of a DB2 data sharing group equals the scope of the OAMplex.

OAM Storage Management Component (OSMC). Where objects should be stored, manages object movement within the object storage hierarchy, and manages expiration attributes based on the installation storage management policy.

OAM thread isolation support (OTIS). An OAM subsystem providing OAM-DB2 functions that use a different thread to DB2 than the application program thread.

object. A named byte stream having no specific format or record orientation.

Object Access Method (OAM). A program that provides object storage, object retrieval, and object storage hierarchy management. OAM isolates applications from storage devices, storage management, and storage device hierarchy management.

Object/Object Backup storage group. A named collection of physical devices to be managed as a single object storage area. It consists of an object directory (DB2 table space), and object storage on DASD (DB2 table spaces) with optional library-resident and shelf-resident optical and or tape volumes.

object directory tables. A collection of DB2 tables that contains information about the objects that have been stored in an SMS Object storage group.

Object Distribution Manager. The application that resides in the image host and provides services to the front-end application hosts for the storage, retrieval, and routing of image objects and coded data.

Object Storage and Retrieval (OSR). Component of OAM that stores, retrieves, and deletes objects. OSR stores objects in the storage hierarchy and maintains the information about these objects in DB2 databases.

object storage database. A DB2 database that contains an object directory for an Object storage group, a storage table for objects less than or equal to 3980 bytes, and a storage table for objects greater than 3980 bytes.

OCDB. Optical configuration database.

Optical Configuration Database (OCDB). The optical library table, the library slot table, the optical drive table, the optical volume table, and the tape volume table that reside in a DB2 database and describe the current OAM configuration.

optical disk. A disk that uses laser technology for data storage and retrieval.

optical disk cartridge. A plastic case that protects and contains the optical disk and permits insertion into an optical drive.

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive may reside in an optical library or as a stand-alone unit.

optical library. A set of optical disk drives and optical disks defined to the source control data set. An optical library can be a real library with the optical drives and optical disks residing within the same storage device, or a pseudo library that consists of operator-accessible drives and shelf-resident optical disks. See also *real optical library, pseudo optical library*.

optical volume. One side of a double-sided optical disk.

OSMC. OAM Storage Management Component.

OSR. Object Storage and Retrieval.

OTIS. OAM thread isolation support.

out-of-space condition.

- A library is considered to be out-of-space for a storage group when:
  - there are no scratch volumes in the optical library
  - any library-resident volumes for the storage group are full.
- The DB2 object database is considered to be out-of-space when:
  - a new row cannot be inserted into the directory entry, or
  - a new row cannot be inserted into the object directory.

OVTOC. Optical Volume Table of Contents.

P

Parallel Sysplex. A sysplex that uses one or more coupling facilities.

PCA. Parallel channel adapter.

PLT. Program list table.

PPT. Program properties table.

primary. An object that is in the object storage hierarchy and can be retrieved by OSREQ RETRIEVE. There is no connection to the last time the object was used or its actual or expected frequency of use.

private. The state of a tape volume that contains user-written data. A private volume is requested by specifying the volume serial number.

private volume. A tape volume that has been assigned the private use attribute by the software. If the cartridge resides in a tape library, it is assigned to the private category.

pseudo optical library. A set of shelf-resident optical volumes associated with stand-alone, or operator-accessible, or both, optical disk drives.

Q

QEL. Query element list.

R

RACF. Resource Access Control Facility.

RCT. Resource control table.

real optical library. Physical storage device that houses optical disk drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives. See also *optical library, pseudo optical library*.

Resource Access Control Facility (RACF). A licensed program that provides computer related security features in an MVS and VM operating system environment.

Resource Control Table (RCT). The CICS table that contains customization information for a particular Object Distribution Manager installation.

resource measurement facility (RMF). A tool to assist in analyzing your business environment if you are currently involved in object support. RMF provides workload and activity reports against specific report performance groups defined in the IEAICSxx PARMLIB member. These reports provide a snapshot status of OAM's performance at a given time.

rewritable media. Media that can be erased, rewritten, or reused.

RMF. Resource measurement facility.

row. The horizontal component of a DB2 table. A row consists of a sequence of values, one for each column of a table.

S

SCDS. Source control data set.

scratch. The state of a tape volume that is available for general use. A scratch volume is requested by omitting the volume serial number.

scratch pool. The collection of tape cartridges from which requests for scratch tapes can be satisfied.

scratch tape. See *scratch volume*.

scratch volume. A tape volume that has been assigned the scratch use attribute by the software. If the cartridge resides in a tape library, it is assigned to a scratch category of the appropriate media type.

SCSI. Small Computer System Interface.

SDR. Sustained data rate.

second backup object. The second backup copy of an object, which is stored in the Object Backup storage group that is specified as a second Object Backup storage group.

sector. On disk storage, an addressable subdivision of a track used to record one block of a program or data.

sequential millisecond response. A parameter specified in the definition of an SMS storage class indicating the desired response time to read the next 4K block of a data entity assuming the prior 4K block has been read.

shelf-resident optical volume. An optical volume that resides outside of an optical library.

slot. A space in a library where a cartridge is stored.

Small computer system interface (SCSI). A mechanical, electrical, and functional standard for a small computer input/output bus and command sets for peripheral device types commonly used with small computers. **Note:** Laser Magnetic Storage International (LMSI) documentation sometimes uses ICI and ISI interchangeably with SCSI.

SMF. System measurement facility.

SMP/E. System Modification Program/Extended.

SMR. Sequential millisecond response.

SMS. Storage Management Subsystem.

SMS complex. A set of up to 32 systems within an installation that are defined to MVS in the base configuration as SMS systems. See also *SMSPLEX*.

SPUFI. SQL Processing Using File Input.

SQL. Structured Query Language.

SQLCODE. Structured query language return code.

SQL Processing Using File Input. Used to perform groups of SQL statements in batch or online mode. SPUFI is option one under DB2I.

stand-alone optical drive. An optical drive housed outside of an optical library.

standard label. An IBM standard tape label.

storage class. A named list of storage attributes. The list of attributes identifies a storage service level provided for data associated with the storage class. No physical storage is directly implied or associated with a given storage class name.

storage group. A named collection of physical devices to be managed as a single object storage area. It consists of an object directory (DB2 table space) and object storage on DASD (DB2 table spaces), with optional library-resident and shelf-resident optical volumes.

storage hierarchy. An arrangement in which data can be stored in several types of storage devices that have different characteristics, such as capacity and speed of access. The types of storage devices include DASD, optical disks, and tape volumes.

storage management cycle. An invocation of the OAM Storage Management Component (OSMC). The purpose of the storage management cycle is to ensure that every object scheduled for processing is placed in the correct level of the object storage hierarchy (as specified by its storage class), is expired or backed up (as specified by its management class or by an explicit application request), and, if necessary, is flagged for action during a subsequent storage management cycle.

Storage Management Subsystem (SMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

structured query language (SQL). A DB2 query tool.

supervisor call (SVC). A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

sustained data rate (SDR). A parameter specified in the definition of an SMS storage class indicating the desired sustained data rate to read the next 4K block of a data entity assuming the prior 4K block has been read.

SVC. Supervisor call.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system measurement facility (SMF). An optional control program feature that provides the means for gathering and recording information that can be used to evaluate system usage.

System Modification Program/Extended (SMP/E). Basic tool for installing software changes in programming systems. It controls these changes at the element (module or macro) level, which helps protect system integrity.

T

table. In DB2, a named data object consisting of a specific number of columns and some number of unordered rows.

table space. A page set used to store the records of one or more DB2 tables.

tape configuration database (TCDB). The set of tape library records and tape volume records that reside in ICF volume catalogs and describe the current tape library configuration.

tape library. For purposes of this support, a set of related tape drives and the set of tape volumes which may be mounted on those drives.

TB. Terabyte.

TCDB. Tape configuration database.

Terabyte (TB). Terabyte as used in data processing (1 099 511 627 776 bytes).

Time Sharing Option (TSO). An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

TSO. Time Sharing Option.

U

user exit. A programming service provided by an IBM software product that may be requested by an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

V

vary offline. To change the status of an optical library or an optical drive from online to offline. Varying a library offline does not affect the online/offline status of the drives it contains. When a library or drive is offline, no data may be accessed on optical disks through the offline drive or the drives in the offline library.

vary online. To change the status of an optical library or an optical drive from offline to online. This makes the drive or drives in the library being varied online available for optical disk access.

volume full threshold. When the number of free kilobytes on the volume falls below this threshold, the volume is marked full.

VTOC. Volume table of contents.

W

WORM. Write-once, read-many.

write-once, read-many (WORM) media. This type of optical disk media cannot be rewritten nor erased.

X

XCF. Cross-system coupling facility.

Index

Numerics

A

D

E

K

L

M

P

Q

R

S

V

X

Readers' Comments — We'd Like to Hear from You

z/OS
DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support

Publication No. SC35-0426-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC35-0426-01

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department 61C
9000 South Rita Road
TUCSON AZ 85775-4401

Fold and Tape | **Please do not staple** | Fold and Tape

SC35-0426-01

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC35-0426-01